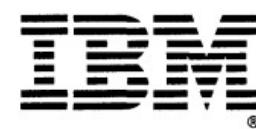


**z/Architecture**



# **Reference Summary**

**z/Architecture**



# **Reference Summary**

## **Third Edition (September, 2005)**

This revision differs from the previous edition by containing instructions related to the facilities marked by a bar under "Facility" in "Preface," fields related to new I/O facilities, and minor corrections and clarifications. Changes are indicated by a bar in the margin.

References in this publication to IBM® products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

**Requests for copies of this and other IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.**

Please direct any comments on the contents of this publication to:

IBM Corporation  
Department E57  
2455 South Road  
Poughkeepsie, NY  
12601-5400  
USA

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

**© Copyright International Business Machines Corporation 2001-2005. All rights reserved.**

US Government Users Restricted Rights — Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

## Preface

This publication is intended primarily for use by z/Architecture™ assembler-language application programmers. It contains basic machine information summarized from the *IBM z/Architecture Principles of Operation*, SA22-7832, about the zSeries™ processors. It also contains frequently used information from *IBM ESA/390 Common I/O-Device Commands and Self Description*, SA22-7204, *IBM System/370 Extended Architecture Interpretive Execution*, SA22-7095, and *IBM High Level Assembler for MVS & VM & VSE Language Reference*, SC26-4940. This publication will be updated from time to time. However, the above publications and others cited in this publication are the authoritative reference sources and will be first to reflect changes.

The following instructions may be uninstalled or not available on a particular model:

Facility	Instruction
ASN and LX reuse	EPAIR, ESAIR, PTI, SSAIR
DAT enhancement	CSPG, IDTE
DAT enhancement 2	LPTEA
Expanded storage	PGIN, PGOUT
Extended immediate	AFI, AGFI, ALFI, ALGFI, CFI, CGFI, CLFI, CLGFI, FLOGR, IIHF, IILF, LBR, LGBR, LGHR, LGFI, LHR, LLC, LLCR, LLGCR, LLGHR, LLH, LLHR, LLIHF, LLILF, LT, LTG, NIHF, NILF, OIHF, OILF, SLFI, SLGFI, XIHF, XILF
Extended translation 2	CLCLU, MVCLU, PKA, PKU, TP, TROO, TROT, TRTO, TRTT, UNPKA, UNPKU
Extended translation 3	CU14, CU24, CU41, CU42, SRSTU, TRTR
HFP multiply-and-add/subtract	MAD, MADR, MAE, MAER, MSD, MSDR, MSE, MSER
HFP unnormalized extensions	MAY, MAYR, MAYH, MAYHR, MAYL, MAYLR, MY, MYH, MYL, MYR, MYHR, MYLR
Long displacement	AHY, ALY, AY, CDSY, CHY, CLIY, CLMY, CLY, CSY, CVBY, CVDY, CY, ICY, ICY, LAMY, LAY, LB, LDY, LEY, LGB, LHY, LMY, LRAY, LY, MSY, MVIY, NIY, NY, OIY, OY, SHY, SLY, STAMY, STCMY, STCY, STDY, STEY, STHY, STMY, STY, SY, TMY, XIY, XY
Message-security assist	KM, KMC, KIMD, KLMD, KMAC
Store-clock fast	STCKF
Store-facility-list extended	STFLE
TOD-clock steering	PTFF

For information about Enterprise Systems Architecture/390® (ESA/390™) architecture, refer to *IBM Enterprise Systems Architecture/390 Principles of Operation*, SA22-7201, and *IBM Enterprise Systems Architecture/390 Reference Summary*, SA22-7209.

**Note:** IBM, z/Architecture, zSeries, Enterprise Systems Architecture/390, and ESA/390 are trademarks of the International Business Machines Corporation in the United States, other countries, or both.



## Contents

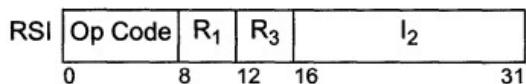
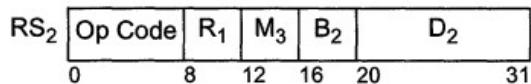
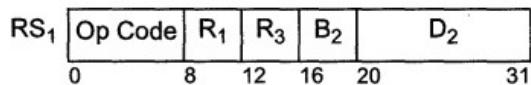
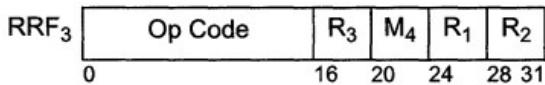
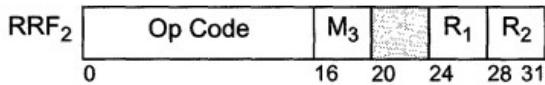
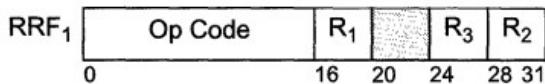
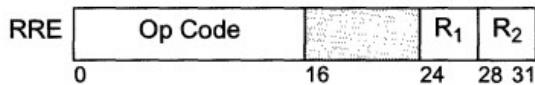
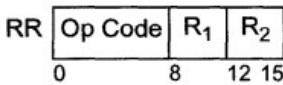
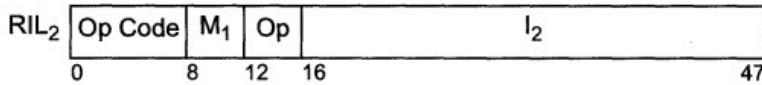
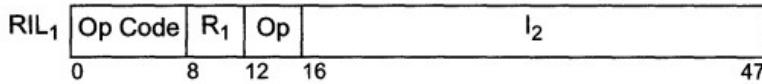
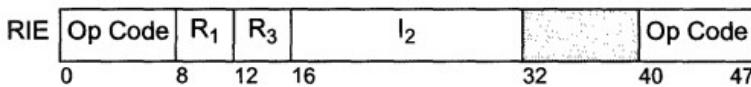
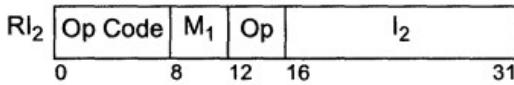
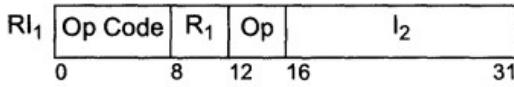
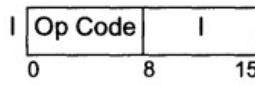
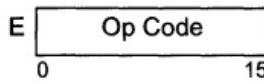
Preface .....	iii
Contents .....	v
Machine Instruction Formats .....	1
Machine Instructions by Mnemonic .....	3
Machine Instructions by Operation Code .....	17
Condition Codes .....	21
Assembler Instructions .....	25
Extended-Mnemonic Instructions for Branch on Condition .....	26
Extended-Mnemonic Instructions for	
Relative-Branch Instructions .....	26
CNOP Alignment .....	27
Summary of Constants .....	28
Operand of Store Clock .....	28
Operand of Store Clock Extended .....	28
Fixed Storage Locations .....	29
External-Interruption Codes .....	30
Program-Interruption Codes .....	30
Translation-Exception Identification .....	31
Data-Exception Code (DXC) .....	32
Facility Indications .....	32
Control Registers .....	33
Floating-Point-Control (FPC) Register .....	35
Program-Status Word (PSW) .....	35
z/Architecture PSW .....	35
ESA/390 PSW .....	36
Dynamic Address Translation .....	36
Virtual-Address Format .....	36
Address-Space-Control Element (ASCE) .....	36
Region-Table or Segment-Table Designation	
(RTD or STD) .....	36
Real-Space Designation (RSD) .....	36
Table Values .....	37
Region-Table Entry (RTE) .....	37
Segment-Table Entry (STE) .....	37
Page-Table Entry (PTE) .....	37
ASN Translation .....	38
Address-Space Number (ASN) .....	38
ASN-First-Table Entry .....	38
ASN-Second-Table Entry (ASTE) .....	38
PC-Number Translation .....	39
Program-Call Number (20-Bit) .....	39
Program-Call Number (32-Bit, Bit 44=0) .....	39
Program-Call Number (32-Bit, Bit 44=1) .....	39
Linkage-Table Entry (LTE) .....	39
Linkage-First-Table Entry (LFTE) .....	39
Linkage-Second-Table Entry (LSTE) .....	40
Entry-Table Entry (ETE) .....	40
Access-Register Translation .....	41
Access-List-Entry Token (ALET) .....	41
Dispatchable-Unit-Control Table (DUCT) .....	41
Access-List Entry (ALE) .....	42
Linkage-Stack Entries .....	42

Entry Descriptor . . . . .	42
Header Entry (Entry Type 0001001) . . . . .	43
Trailer Entry (Entry Type 0001010) . . . . .	43
Branch State Entry (Entry Type 0001100) and Program-Call State Entry (Entry Type 0001101) . . . . .	43
Trapping . . . . .	44
Trap Control Block . . . . .	44
Trap Save Area . . . . .	45
Trace-Entry Formats . . . . .	46
Identification of Trace Entries . . . . .	46
Branch . . . . .	47
Branch in Subspace Group (if ASN Tracing on) . . . . .	47
Mode Switch . . . . .	47
Mode-Switching Branch . . . . .	48
Program Call . . . . .	48
Program Return . . . . .	49
Program Transfer . . . . .	50
Set Secondary ASN . . . . .	51
Trace . . . . .	51
Machine-Check Interruption Code . . . . .	52
External-Damage Code . . . . .	52
Operation-Request Block (ORB) . . . . .	53
Channel-Command Word (CCW) . . . . .	53
Format-0 CCW . . . . .	53
Format-1 CCW . . . . .	54
Indirect-Data-Address Word (IDAW) . . . . .	54
Format-1 IDAW . . . . .	54
Format-2 IDAW . . . . .	54
Modified-CCW-Indirect-Data-Address Word (MIDAW) . . . . .	54
Subchannel-Information Block (SCHIB) . . . . .	55
Path-Management-Control Word (PMCW) . . . . .	55
Interruption-Response Block (IRB) . . . . .	56
Subchannel-Status Word (SCSW) . . . . .	56
Extended-Status Word (ESW) . . . . .	57
Format-0 ESW . . . . .	57
Format-0 ESW Word 0 (Subchannel Logout) . . . . .	57
Format-0 ESW Word 1 (Extended-Report Word) . . . . .	58
Format-1 ESW Word 0 . . . . .	58
Format-2 ESW Word 0 <sup>1</sup> . . . . .	58
Format-3 ESW Word 0 <sup>1</sup> . . . . .	58
Information Stored in ESW . . . . .	58
Extended-Control Word (ECW) . . . . .	59
Extended-Measurement Word . . . . .	60
Format 0 Measurement Block . . . . .	60
Format 1 Measurement Block . . . . .	60
Channel-Report Word (CRW) . . . . .	61
Error-Recovery Codes . . . . .	61
Reporting Source . . . . .	61
I/O Command Codes . . . . .	61
Standard Command-Code Assignments (CCW Bits 0-7) . . . . .	61
Standard Meanings of Bits of First Sense Byte . . . . .	61
Code Assignments . . . . .	62
Control Character Representations . . . . .	64

Additional ISO-8 Control Character Representations	64
Formatting Character Representations	64
Two-Character BSC Data Link Controls	64
Commonly Used Editing Pattern Characters	64
ANSI-Defined Printer Control Characters	65
Hexadecimal and Decimal Conversion	65
Powers of 2 and 16	67



## Machine Instruction Formats



RSL	<table border="1"> <tr> <td>Op Code</td><td>L<sub>1</sub></td><td></td><td>B<sub>2</sub></td><td></td><td>D<sub>2</sub></td><td></td><td>Op Code</td></tr> </table>	Op Code	L <sub>1</sub>		B <sub>2</sub>		D <sub>2</sub>		Op Code
Op Code	L <sub>1</sub>		B <sub>2</sub>		D <sub>2</sub>		Op Code		
	0      8      12      16      20      32      40      47								

RSY <sub>1</sub>	<table border="1"> <tr> <td>Op Code</td><td>R<sub>1</sub></td><td>R<sub>3</sub></td><td>B<sub>2</sub></td><td></td><td>DL<sub>2</sub></td><td>DH<sub>2</sub></td><td>Op Code</td></tr> </table>	Op Code	R <sub>1</sub>	R <sub>3</sub>	B <sub>2</sub>		DL <sub>2</sub>	DH <sub>2</sub>	Op Code
Op Code	R <sub>1</sub>	R <sub>3</sub>	B <sub>2</sub>		DL <sub>2</sub>	DH <sub>2</sub>	Op Code		
	0      8      12      16      20      32      40      47								

RSY <sub>2</sub>	<table border="1"> <tr> <td>Op Code</td><td>R<sub>1</sub></td><td>M<sub>3</sub></td><td>B<sub>2</sub></td><td></td><td>DL<sub>2</sub></td><td>DH<sub>2</sub></td><td>Op Code</td></tr> </table>	Op Code	R <sub>1</sub>	M <sub>3</sub>	B <sub>2</sub>		DL <sub>2</sub>	DH <sub>2</sub>	Op Code
Op Code	R <sub>1</sub>	M <sub>3</sub>	B <sub>2</sub>		DL <sub>2</sub>	DH <sub>2</sub>	Op Code		
	0      8      12      16      20      32      40      47								

RX	<table border="1"> <tr> <td>Op Code</td><td>R<sub>1</sub></td><td>X<sub>2</sub></td><td>B<sub>2</sub></td><td></td><td>D<sub>2</sub></td><td></td><td></td></tr> </table>	Op Code	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>		
Op Code	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>				
	0      8      12      16      20      31								

RXE	<table border="1"> <tr> <td>Op Code</td><td>R<sub>1</sub></td><td>X<sub>2</sub></td><td>B<sub>2</sub></td><td></td><td>D<sub>2</sub></td><td></td><td>Op Code</td></tr> </table>	Op Code	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>		Op Code
Op Code	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		D <sub>2</sub>		Op Code		
	0      8      12      16      20      32      40      47								

RXY	<table border="1"> <tr> <td>Op Code</td><td>R<sub>1</sub></td><td>X<sub>2</sub></td><td>B<sub>2</sub></td><td></td><td>DL<sub>2</sub></td><td>DH<sub>2</sub></td><td>Op Code</td></tr> </table>	Op Code	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		DL <sub>2</sub>	DH <sub>2</sub>	Op Code
Op Code	R <sub>1</sub>	X <sub>2</sub>	B <sub>2</sub>		DL <sub>2</sub>	DH <sub>2</sub>	Op Code		
	0      8      12      16      20      32      40      47								

S	<table border="1"> <tr> <td>Op Code</td><td></td><td>B<sub>1</sub></td><td></td><td>D<sub>1</sub></td><td></td><td></td></tr> </table>	Op Code		B <sub>1</sub>		D <sub>1</sub>		
Op Code		B <sub>1</sub>		D <sub>1</sub>				
	0      16      20      31							

SI	<table border="1"> <tr> <td>Op Code</td><td>I<sub>2</sub></td><td></td><td>B<sub>1</sub></td><td></td><td>D<sub>1</sub></td><td></td><td></td></tr> </table>	Op Code	I <sub>2</sub>		B <sub>1</sub>		D <sub>1</sub>		
Op Code	I <sub>2</sub>		B <sub>1</sub>		D <sub>1</sub>				
	0      8      16      20      31								

SIY	<table border="1"> <tr> <td>Op Code</td><td>I<sub>2</sub></td><td></td><td>B<sub>1</sub></td><td></td><td>DL<sub>1</sub></td><td>DH<sub>1</sub></td><td>Op Code</td></tr> </table>	Op Code	I <sub>2</sub>		B <sub>1</sub>		DL <sub>1</sub>	DH <sub>1</sub>	Op Code
Op Code	I <sub>2</sub>		B <sub>1</sub>		DL <sub>1</sub>	DH <sub>1</sub>	Op Code		
	0      8      16      20      32      40      47								

SS <sub>1</sub>	<table border="1"> <tr> <td>Op Code</td><td>L</td><td></td><td>B<sub>1</sub></td><td></td><td>D<sub>1</sub></td><td>B<sub>2</sub></td><td>D<sub>2</sub></td><td></td></tr> </table>	Op Code	L		B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>	
Op Code	L		B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>			
	0      8      16      20      32      36      47									

SS <sub>2</sub>	<table border="1"> <tr> <td>Op Code</td><td>L<sub>1</sub></td><td>L<sub>3</sub></td><td>B<sub>1</sub></td><td></td><td>D<sub>1</sub></td><td>B<sub>2</sub></td><td>D<sub>2</sub></td><td></td></tr> </table>	Op Code	L <sub>1</sub>	L <sub>3</sub>	B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>	
Op Code	L <sub>1</sub>	L <sub>3</sub>	B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>			
	0      8      12      16      20      32      36      47									

SS <sub>3</sub>	<table border="1"> <tr> <td>Op Code</td><td>L<sub>1</sub></td><td>I<sub>3</sub></td><td>B<sub>1</sub></td><td></td><td>D<sub>1</sub></td><td>B<sub>2</sub></td><td>D<sub>2</sub></td><td></td></tr> </table>	Op Code	L <sub>1</sub>	I <sub>3</sub>	B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>	
Op Code	L <sub>1</sub>	I <sub>3</sub>	B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>			
	0      8      12      16      20      32      36      47									

SS <sub>4</sub>	<table border="1"> <tr> <td>Op Code</td><td>R<sub>1</sub></td><td>R<sub>3</sub></td><td>B<sub>1</sub></td><td></td><td>D<sub>1</sub></td><td>B<sub>2</sub></td><td>D<sub>2</sub></td><td></td></tr> </table>	Op Code	R <sub>1</sub>	R <sub>3</sub>	B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>	
Op Code	R <sub>1</sub>	R <sub>3</sub>	B <sub>1</sub>		D <sub>1</sub>	B <sub>2</sub>	D <sub>2</sub>			
	0      8      12      16      20      32      36      47									

SS <sub>5</sub>	<table border="1"> <tr> <td>Op Code</td><td>R<sub>1</sub></td><td>R<sub>3</sub></td><td>B<sub>2</sub></td><td></td><td>D<sub>2</sub></td><td>B<sub>4</sub></td><td>D<sub>4</sub></td><td></td></tr> </table>	Op Code	R <sub>1</sub>	R <sub>3</sub>	B <sub>2</sub>		D <sub>2</sub>	B <sub>4</sub>	D <sub>4</sub>	
Op Code	R <sub>1</sub>	R <sub>3</sub>	B <sub>2</sub>		D <sub>2</sub>	B <sub>4</sub>	D <sub>4</sub>			
	0      8      12      16      20      32      36      47									

SSE	<table border="1"> <tr> <td>Op Code</td><td></td><td>B<sub>1</sub></td><td></td><td>D<sub>1</sub></td><td></td><td>B<sub>2</sub></td><td></td><td>D<sub>2</sub></td></tr> </table>	Op Code		B <sub>1</sub>		D <sub>1</sub>		B <sub>2</sub>		D <sub>2</sub>
Op Code		B <sub>1</sub>		D <sub>1</sub>		B <sub>2</sub>		D <sub>2</sub>		
	0      16      20      32      36      47									

1, 2, 3, 4:	Denotes association with first, second, third, or fourth operand; distinguishes among multiple instances of the same instruction format
B <sub>1</sub> , B <sub>2</sub> , B <sub>4</sub> :	Base register designation field
D <sub>1</sub> , D <sub>2</sub> , D <sub>4</sub> :	Displacement field
I, I <sub>2</sub> , I <sub>3</sub> :	Immediate operand field
L, L <sub>1</sub> , L <sub>2</sub> :	Length field
M <sub>1</sub> , M <sub>3</sub> , M <sub>4</sub> :	Mask field
R <sub>1</sub> , R <sub>2</sub> , R <sub>3</sub> :	Register designation field
X <sub>2</sub> :	Index register designation field

## Machine Instructions by Mnemonic

Mne-monics	Operands	Name	For-mat	Op-code	Class & Notes
A	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add (32)	RX	5A	c
AD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Normalized (LH)	RX	6A	c
ADB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add (LB)	RXE	ED1A	c
ADBR	R <sub>1</sub> ,R <sub>2</sub>	Add (LB)	RRE	B31A	c
ADR	R <sub>1</sub> ,R <sub>2</sub>	Add Normalized (LH)	RR	2A	c
AE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Normalized (SH)	RX	7A	c
AEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add (SB)	RXE	ED0A	c
AEBR	R <sub>1</sub> ,R <sub>2</sub>	Add (SB)	RRE	B30A	c
AER	R <sub>1</sub> ,R <sub>2</sub>	Add Normalized (SH)	RR	3A	c
AFI	R <sub>1</sub> ,I <sub>2</sub>	Add Immediate (32)	RIL	C29	c EI
AG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add (64)	RXY	E308	c N
AGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add (64<32)	RXY	E318	c N
AGFI	R <sub>1</sub> ,I <sub>2</sub>	Add Immediate (64<32)	RIL	C28	c EI
AGFR	R <sub>1</sub> ,R <sub>2</sub>	Add (64<32)	RRE	B918	c N
AGHI	R <sub>1</sub> ,I <sub>2</sub>	Add Halfword Immediate	RI <sub>1</sub>	A7B	c N
AGR	R <sub>1</sub> ,R <sub>2</sub>	Add (64)	RRE	B908	c N
AH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Halfword	RX	4A	c
AHI	R <sub>1</sub> ,I <sub>2</sub>	Add Halfword Immediate (32)	RI <sub>1</sub>	A7A	c
AHY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Halfword	RXY	E37A	c LD
AL	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Logical (32)	RX	5E	c
ALC	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Logical with Carry (32)	RXY	E398	c N3
ALCG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Logical with Carry (64)	RXY	E388	c N
ALCGR	R <sub>1</sub> ,R <sub>2</sub>	Add Logical with Carry (64)	RRE	B988	c N
ALCR	R <sub>1</sub> ,R <sub>2</sub>	Add Logical with Carry (32)	RRE	B998	c N3
ALFI	R <sub>1</sub> ,I <sub>2</sub>	Add Logical Immediate (32)	RIL	C2B	c EI
ALG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Logical (64)	RXY	E30A	c N
ALGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Logical (64<32)	RXY	E31A	c N
ALGFI	R <sub>1</sub> ,I <sub>2</sub>	Add Logical Immediate (64<32)	RIL	C2A	c EI
ALGFR	R <sub>1</sub> ,R <sub>2</sub>	Add Logical (64<32)	RRE	B91A	c N
ALGR	R <sub>1</sub> ,R <sub>2</sub>	Add Logical (64)	RRE	B90A	c N
ALR	R <sub>1</sub> ,R <sub>2</sub>	Add Logical (32)	RR	1E	c
ALY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Logical (32)	RXY	E35E	c LD
AP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Add Decimal	SS <sub>2</sub>	FA	c
AR	R <sub>1</sub> ,R <sub>2</sub>	Add (32)	RR	1A	c
AU	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Unnormalized (SH)	RX	7E	c
AUR	R <sub>1</sub> ,R <sub>2</sub>	Add Unnormalized (SH)	RR	3E	c
AW	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add Unnormalized (LH)	RX	6E	c
AWR	R <sub>1</sub> ,R <sub>2</sub>	Add Unnormalized (LH)	RR	2E	c
AXB	R <sub>1</sub> ,R <sub>2</sub>	Add (EB)	RRE	B34A	c
AXR	R <sub>1</sub> ,R <sub>2</sub>	Add Normalized (EH)	RR	36	c
AY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Add (32)	RXY	E35A	c LD
BAKR	R <sub>1</sub> ,R <sub>2</sub>	Branch and Stack	RRE	B240	q
BAL	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Branch and Link	RX	45	

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
BALR	R <sub>1</sub> ,R <sub>2</sub>	Branch and Link	RR	05	
BAS	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Branch and Save	RX	4D	
BASR	R <sub>1</sub> ,R <sub>2</sub>	Branch and Save	RR	0D	
BASSM	R <sub>1</sub> ,R <sub>2</sub>	Branch and Save and Set Mode	RR	0C	
BC	M <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Branch on Condition	RX	47	
BCR	M <sub>1</sub> ,R <sub>2</sub>	Branch on Condition	RR	07	
BCT	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Branch on Count (32)	RX	46	
BCTG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Branch on Count (64)	RXY	E346	N
BCTGR	R <sub>1</sub> ,R <sub>2</sub>	Branch on Count (64)	RRE	B946	N
BCTR	R <sub>1</sub> ,R <sub>2</sub>	Branch on Count (32)	RR	06	
BRAS	R <sub>1</sub> ,I <sub>2</sub>	Branch Relative and Save	RI <sub>1</sub>	A75	
BRASL	R <sub>1</sub> ,I <sub>2</sub>	Branch Relative and Save Long	RIL <sub>1</sub>	C05	N3
BRC	M <sub>1</sub> ,I <sub>2</sub>	Branch Relative on Condition	RI <sub>2</sub>	A74	
BRCL	M <sub>1</sub> ,I <sub>2</sub>	Branch Relative on Condition Long	RIL <sub>2</sub>	C04	N3
BRCT	R <sub>1</sub> ,I <sub>2</sub>	Branch Relative on Count (32)	RI <sub>1</sub>	A76	
BRCTG	R <sub>1</sub> ,I <sub>2</sub>	Branch Relative on Count (64)	RI <sub>1</sub>	A77	N
BRXH	R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>	Branch Relative on Index High (32)	RSI	84	
BRXHG	R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>	Branch Relative on Index High (64)	RIE	EC44	N
BRXLE	R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>	Branch Relative on Index Low or Equal (32)	RSI	85	
BRXLG	R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>	Branch Relative on Index Low or Equal (64)	RIE	EC45	N
BSA	R <sub>1</sub> ,R <sub>2</sub>	Branch and Set Authority	RRE	B25A	q
BSG	R <sub>1</sub> ,R <sub>2</sub>	Branch in Subspace Group	RRE	B258	
BSM	R <sub>1</sub> ,R <sub>2</sub>	Branch and Set Mode	RR	0B	
BXH	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Branch on Index High (32)	RS <sub>1</sub>	86	
BXHG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Branch on Index High (64)	RSY <sub>1</sub>	EB44	N
BXLE	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Branch on Index Low or Equal (32)	RS <sub>1</sub>	87	
BXLEG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Branch on Index Low or Equal (64)	RSY <sub>1</sub>	EB45	N
C	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (32)	RX	59	c
CD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (LH)	RX	69	c
CDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (LB)	RXE	ED19	c
CDBR	R <sub>1</sub> ,R <sub>2</sub>	Compare (LB)	RRE	B319	c
CDFBR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (LB<32)	RRE	B395	
CDFR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (LH<32)	RRE	B3B5	
CDGBR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (LB<64)	RRE	B3A5	N
CDGR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (LB<64)	RRE	B3C5	N
CDR	R <sub>1</sub> ,R <sub>2</sub>	Compare (LH)	RR	29	c
CDS	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Double and Swap (32)	RS <sub>1</sub>	BB	c
CDSG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Double and Swap (64)	RSY <sub>1</sub>	EB3E	c N
CDSY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Double and Swap (32)	RSY <sub>1</sub>	EB31	c LD
CE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (SH)	RX	79	c
CEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (SB)	RXE	ED09	c
CEBR	R <sub>1</sub> ,R <sub>2</sub>	Compare (SB)	RRE	B309	c
CEFBR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (SB<32)	RRE	B394	
CEFR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (SH<32)	RRE	B3B4	
CEGBR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (SB<64)	RRE	B3A4	N
CEGR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (SH<64)	RRE	B3C4	N
CER	R <sub>1</sub> ,R <sub>2</sub>	Compare (SH)	RR	39	c
CFC	D <sub>2</sub> (B <sub>2</sub> )	Compare and Form Codeword	S	B21A	ic
CFDBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (32<LB)	RRF <sub>2</sub>	B399	c

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
CFDR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (32<LH)	RRF <sub>2</sub>	B3B9	c
CFEBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (32<SB)	RRF <sub>2</sub>	B398	c
CFER	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (32<SH)	RRF <sub>2</sub>	B3B8	c
CFI	R <sub>1</sub> ,I <sub>2</sub>	Compare Immediate (32)	RIL	C2D	c EI
CFXBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (32<EB)	RRF <sub>2</sub>	B39A	c
CFXR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (32<EH)	RRF <sub>2</sub>	B3BA	c
CG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (64)	RXY	E320	c N
CGDBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (64<LB)	RRF <sub>2</sub>	B3A9	c N
CGDR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (64<LH)	RRF <sub>2</sub>	B3C9	c N
CGEBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (64<SB)	RRF <sub>2</sub>	B3A8	c N
CGER	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (64<SH)	RRF <sub>2</sub>	B3C8	c N
CGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (64<32)	RXY	E330	c N
CGFI	R <sub>1</sub> ,I <sub>2</sub>	Compare Immediate (64<32)	RIL	C2C	c EI
CGFR	R <sub>1</sub> ,R <sub>2</sub>	Compare (64<32)	RRE	B930	c N
CGHI	R <sub>1</sub> ,I <sub>2</sub>	Compare Halfword Immediate (64)	RI <sub>1</sub>	A7F	c N
CGR	R <sub>1</sub> ,R <sub>2</sub>	Compare (64)	RRE	B920	c N
CGXBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (64<EB)	RRF <sub>2</sub>	B3AA	c N
CGXR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert to Fixed (64<EH)	RRF <sub>2</sub>	B3CA	c N
CH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare Halfword	RX	49	c
CHI	R <sub>1</sub> ,I <sub>2</sub>	Compare Halfword Immediate (32)	RI <sub>1</sub>	A7E	c
CHY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare Halfword	RXY	E379	c LD
CKSM	R <sub>1</sub> ,R <sub>2</sub>	Checksum	RRE	B241	c
CL	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare Logical (32)	RX	55	c
CLC	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Compare Logical (character)	SS <sub>1</sub>	D5	c
CLCL	R <sub>1</sub> ,R <sub>2</sub>	Compare Logical Long	RR	0F	ic
CLCLE	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Logical Long Extended	RS <sub>1</sub>	A9	c
CLCLU	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Logical Long Unicode	RSY <sub>1</sub>	EB8F	c E2
CLFI	R <sub>1</sub> ,I <sub>2</sub>	Compare Logical Immediate (32)	RIL	C2F	c EI
CLG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare Logical (64)	RXY	E321	c N
CLGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare Logical (64<32)	RXY	E331	c N
CLGFI	R <sub>1</sub> ,I <sub>2</sub>	Compare Logical Immediate (64<32)	RIL	C2E	c EI
CLGFR	R <sub>1</sub> ,R <sub>2</sub>	Compare Logical (64<32)	RRE	B931	c N
CLGR	R <sub>1</sub> ,R <sub>2</sub>	Compare Logical (64)	RRE	B921	c N
CLI	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Compare Logical (immediate)	SI	95	c
CLIY	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Compare Logical (immediate)	SIY	EB55	c LD
CLM	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Logical Characters under Mask	RS <sub>2</sub>	BD	c
CLMH	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Logical Characters under Mask	RSY <sub>2</sub>	EB20	c N
CLMY	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare Logical Characters under Mask	RSY <sub>2</sub>	EB21	c LD
CLR	R <sub>1</sub> ,R <sub>2</sub>	Compare Logical (32)	RR	15	c
CLST	R <sub>1</sub> ,R <sub>2</sub>	Compare Logical String	RRE	B25D	c
CLY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare Logical (32)	RXY	E355	c LD
CMPSC	R <sub>1</sub> ,R <sub>2</sub>	Compression Call	RRE	B263	ic
CP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Compare Decimal	SS <sub>2</sub>	F9	c
CPYA	R <sub>1</sub> ,R <sub>2</sub>	Copy Access	RRE	B24D	
CR	R <sub>1</sub> ,R <sub>2</sub>	Compare (32)	RR	19	c
CS	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare and Swap (32)	RS <sub>1</sub>	BA	c
CSCH		Clear Subchannel	S	B230	pc
CSG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare and Swap (64)	RSY <sub>1</sub>	EB30	c N
CSP	R <sub>1</sub> ,R <sub>2</sub>	Compare and Swap and Purge (32)	RRE	B250	pc

Mne-monics	Operands	Name	For-mat	Op-code	Class & Notes
CSPG	R <sub>1</sub> ,R <sub>2</sub>	Compare and Swap and Purge (64)	RRE	B98A	pc DE
CSY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Compare and Swap (32)	RSY	EB14	c LD
CU12	R <sub>1</sub> ,R <sub>2</sub> [M <sub>3</sub> ]	Convert UTF-8 to UTF-16	RRF	B2A7	c
CU14	R <sub>1</sub> ,R <sub>2</sub> [M <sub>3</sub> ]	Convert UTF-8 to UTF-32	RRF	B9B0	c E3
CU21	R <sub>1</sub> ,R <sub>2</sub> [M <sub>3</sub> ]	Convert UTF-16 to UTF-8	RRF	B2A6	c
CU24	R <sub>1</sub> ,R <sub>2</sub> [M <sub>3</sub> ]	Convert UTF-16 to UTF-32	RRF	B9B1	c E3
CU41	R <sub>1</sub> ,R <sub>2</sub>	Convert UTF-32 to UTF-8	RRE	B9B2	c E3
CU42	R <sub>1</sub> ,R <sub>2</sub>	Convert UTF-32 to UTF-32	RRE	B9B3	c E3
CUSE	R <sub>1</sub> ,R <sub>2</sub>	Compare until Substring Equal	RRE	B257	ic
CUTFU	R <sub>1</sub> ,R <sub>2</sub> [M <sub>3</sub> ]	Convert UTF-8 to Unicode	RRF	B2A7	c
CUUTF	R <sub>1</sub> ,R <sub>2</sub> [M <sub>3</sub> ]	Convert Unicode to UTF-8	RRF	B2A6	c
CVB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Convert to Binary (32)	RX	4F	
CVBG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Convert to Binary (64)	RXY	E30E	N
CVBY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Convert to Binary (32)	RXY	EB06	LD
CVD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Convert to Decimal (32)	RX	4E	
CVDG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Convert to Decimal (64)	RXY	E32E	N
CVDY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Convert to Decimal (32)	RXY	E326	LD
CXBR	R <sub>1</sub> ,R <sub>2</sub>	Compare (EB)	RRE	B349	c
CXFBR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (EB<32)	RRE	B396	
CXFR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (EH<32)	RRE	B3B6	
CXGBR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (EB<64)	RRE	B3A6	N
CXGR	R <sub>1</sub> ,R <sub>2</sub>	Convert from Fixed (EH<64)	RRE	B3C6	N
CXR	R <sub>1</sub> ,R <sub>2</sub>	Compare (EH)	RRE	B369	c
CY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare (32)	RXY	E359	c LD
D	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide (32<64)	RX	5D	
DD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide (LH)	RX	6D	
DBB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide (LB)	RXE	ED1D	
DDBR	R <sub>1</sub> ,R <sub>2</sub>	Divide (LB)	RRE	B31D	
DDR	R <sub>1</sub> ,R <sub>2</sub>	Divide (LH)	RR	2D	
DE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide (SH)	RX	7D	
DEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide (SB)	RXE	ED0D	
DEBR	R <sub>1</sub> ,R <sub>2</sub>	Divide (SB)	RRE	B30D	
DER	R <sub>1</sub> ,R <sub>2</sub>	Divide (SH)	RR	3D	
DIDBR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub> ,M <sub>4</sub>	Divide to Integer (LB)	RRF	B35B	c
DIEBR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub> ,M <sub>4</sub>	Divide to Integer (SB)	RRF	B353	c
DL	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide Logical (32<64)	RXY	E397	N3
DLG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide Logical (64<128)	RXY	E387	N
DLGR	R <sub>1</sub> ,R <sub>2</sub>	Divide Logical (64<128)	RRE	B987	N
DLR	R <sub>1</sub> ,R <sub>2</sub>	Divide Logical (32<64)	RRE	B997	N3
DP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Divide Decimal	SS <sub>2</sub>	FD	
DR	R <sub>1</sub> ,R <sub>2</sub>	Divide	RR	1D	
DSG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide Single (64)	RXY	E30D	N
DSGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Divide Single (64<32)	RXY	E31D	N
DSGFR	R <sub>1</sub> ,R <sub>2</sub>	Divide Single (64<32)	RRE	B91D	N
DSGR	R <sub>1</sub> ,R <sub>2</sub>	Divide Single (64)	RRE	B90D	N
DXBR	R <sub>1</sub> ,R <sub>2</sub>	Divide (EB)	RRE	B34D	
DXR	R <sub>1</sub> ,R <sub>2</sub>	Divide (EH)	RRE	B22D	
EAR	R <sub>1</sub> ,R <sub>2</sub>	Extract Access	RRE	B24F	
ED	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Edit	SS <sub>1</sub>	DE	c
EDMK	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Edit and Mark	SS <sub>1</sub>	DF	c
EFPC	R <sub>1</sub>	Extract FPC	RRE	B38C	
EPAIR	R <sub>1</sub>	Extract Primary ASN and Instance	RRE	B99A	q RA
EPAR	R <sub>1</sub>	Extract Primary ASN	RRE	B226	q
EPSW	R <sub>1</sub> ,R <sub>2</sub>	Extract PSW	RRE	B98D	N3
EREG	R <sub>1</sub> ,R <sub>2</sub>	Extract Stacked Registers (32)	RRE	B249	

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
EREGG	R <sub>1</sub> ,R <sub>2</sub>	Extract Stacked Registers (64)	RRE	B90E	N
ESAIR	R <sub>1</sub>	Extract Secondary ASN and Instance	RRE	B99B	q RA
ESAR	R <sub>1</sub>	Extract Secondary ASN	RRE	B227	q
ESEA	R <sub>1</sub> ,R <sub>2</sub>	Extract and Set Extended Authority	RRE	B99D	p N
ESTA	R <sub>1</sub> ,R <sub>2</sub>	Extract Stacked State	RRE	B24A	c
EX	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Execute	RX	44	
FIDBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Load FP Integer (LB)	RRF <sub>2</sub>	B35F	
FIDR	R <sub>1</sub> ,R <sub>2</sub>	Load FP Integer (LH)	RRE	B37F	
FIEBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Load FP Integer (SB)	RRF <sub>2</sub>	B357	
FIER	R <sub>1</sub> ,R <sub>2</sub>	Load FP Integer (SH)	RRE	B377	
FIXBR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Load FP Integer (EB)	RRF <sub>2</sub>	B347	
FIXR	R <sub>1</sub> ,R <sub>2</sub>	Load FP Integer (EH)	RRE	B367	
FLOGR	R <sub>1</sub> ,R <sub>2</sub>	Find Leftmost One	RRE	B983	c EI
HDR	R <sub>1</sub> ,R <sub>2</sub>	Halve (LH)	RR	24	
HER	R <sub>1</sub> ,R <sub>2</sub>	Halve (SH)	RR	34	
HSCH		Halt Subchannel	S	B231	pc
IAC	R <sub>1</sub>	Insert Address Space Control	RRE	B224	qc
IC	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Insert Character	RX	43	
ICM	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Insert Characters under Mask (low)	RS <sub>2</sub>	BF	c
ICMH	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Insert Characters under Mask (high)	RSY <sub>2</sub>	EB80	c N
ICMY	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Insert Characters under Mask (low)	RSY <sub>2</sub>	EB81	c LD
ICY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Insert Character	RXY	E373	LD
IDTE	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Invalidate DAT Table Entry	RRF <sub>3</sub>	B98E	pu DE
IIHF	R <sub>1</sub> ,I <sub>2</sub>	Insert Immediate (high)	RIL	C08	EI
IIHH	R <sub>1</sub> ,I <sub>2</sub>	Insert Immediate (high high)	RI <sub>1</sub>	A50	N
IIHL	R <sub>1</sub> ,I <sub>2</sub>	Insert Immediate (high low)	RI <sub>1</sub>	A51	N
IILF	R <sub>1</sub> ,I <sub>2</sub>	Insert Immediate (low)	RIL	C09	EI
IILH	R <sub>1</sub> ,I <sub>2</sub>	Insert Immediate (low high)	RI <sub>1</sub>	A52	N
IILL	R <sub>1</sub> ,I <sub>2</sub>	Insert Immediate (low low)	RI <sub>1</sub>	A53	N
IPK		Insert PSW Key	S	B20B	q
IPTE	R <sub>1</sub> ,R <sub>2</sub>	Invalidate Page Table Entry	RRE	B221	p
ISKE	R <sub>1</sub> ,R <sub>2</sub>	Insert Storage Key Extended	RRE	B229	p
IVSK	R <sub>1</sub> ,R <sub>2</sub>	Insert Virtual Storage Key	RRE	B223	q
KDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare and Signal (LB)	RXE	ED18	c
KDBR	R <sub>1</sub> ,R <sub>2</sub>	Compare and Signal (LB)	RRE	B318	c
KEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Compare and Signal (SB)	RXE	ED08	c
KEBR	R <sub>1</sub> ,R <sub>2</sub>	Compare and Signal (SB)	RRE	B308	c
KIMD	R <sub>1</sub> ,R <sub>2</sub>	Compute Intermediate Message Digest	RRE	B93E	c MS
KLMD	R <sub>1</sub> ,R <sub>2</sub>	Compute Last Message Digest	RRE	B93F	c MS
KM	R <sub>1</sub> ,R <sub>2</sub>	Cipher Message	RRE	B92E	c MS
KMAC	R <sub>1</sub> ,R <sub>2</sub>	Compute Message Authentication Code	RRE	B91E	c MS
KMC	R <sub>1</sub> ,R <sub>2</sub>	Cipher Message with Chaining	RRE	B92F	c MS
KXBR	R <sub>1</sub> ,R <sub>2</sub>	Compare and Signal (EB)	RRE	B348	c
L	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (32)	RX	58	
LA	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Address	RX	41	
LAE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Address Extended	RX	51	
LAM	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Access Multiple	RS <sub>1</sub>	9A	
LAMY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Access Multiple	RSY <sub>1</sub>	EB9A	LD
LARL	R <sub>1</sub> ,I <sub>2</sub>	Load Address Relative Long	RIL <sub>1</sub>	C00	N3
LASP	D <sub>1</sub> (B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Load Address Space Parameters	SSE	E500	pc

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
LAY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Address	RXY	E371	LD
LB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Byte (32)	RXY	E376	LD
LBR	R <sub>1</sub> ,R <sub>2</sub>	Load Byte (32)	RRE	B926	EI
LCDBR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (LB)	RRE	B313	c
LCDR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (LH)	RR	23	c
LCEBR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (SB)	RRE	B303	c
LCER	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (S)	RR	33	c
LCGFR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (64<32)	RRE	B913	c N
LCGR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (64)	RRE	B903	c N
LCR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (32)	RR	13	c
LCTL	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Control (32)	RS <sub>1</sub>	B7	p
LCTLG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Control (64)	RSY <sub>1</sub>	EB2F	p N
LCXBR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (EB)	RRE	B343	c
LCXR	R <sub>1</sub> ,R <sub>2</sub>	Load Complement (EH)	RRE	B363	c
LD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (L)	RX	68	
LDE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Lengthened (LH<SH)	RXE	ED24	
LDEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Lengthened (LB<SB)	RXE	ED04	
LDEBR	R <sub>1</sub> ,R <sub>2</sub>	Load Lengthened (LB<SB)	RRE	B304	
LDER	R <sub>1</sub> ,R <sub>2</sub>	Load Lengthened (LH<SH)	RRE	B324	
LDR	R <sub>1</sub> ,R <sub>2</sub>	Load (L)	RR	28	
LDXBR	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (LB<EB)	RRE	B345	
LDXR	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (LH<EH)	RR	25	
LDY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (L)	RXY	ED65	LD
LE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (S)	RX	78	
LEDBR	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (SB<LB)	RRE	B344	
LEDR	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (SH<LH)	RR	35	
LER	R <sub>1</sub> ,R <sub>2</sub>	Load (S)	RR	38	
LEXBR	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (SB<EB)	RRE	B346	
LEXR	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (SH<EH)	RRE	B366	
LEY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (S)	RXY	ED64	LD
LFPC	D <sub>2</sub> (B <sub>2</sub> )	Load FPC	S	B29D	
LG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (64)	RXY	E304	N
LGB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Byte (64)	RXY	E377	LD
LGBR	R <sub>1</sub> ,R <sub>2</sub>	Load Byte (64)	RRE	B906	EI
LGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (64<32)	RXY	E314	N
LGFI	R <sub>1</sub> ,I <sub>2</sub>	Load Immediate (32<64)	RIL	C01	EI
LGFR	R <sub>1</sub> ,R <sub>2</sub>	Load (64<32)	RRE	B914	N
LGH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Halfword (64)	RXY	E315	N
LGHI	R <sub>1</sub> ,I <sub>2</sub>	Load Halfword Immediate (64)	RI <sub>1</sub>	A79	N
LGHR	R <sub>1</sub> ,R <sub>2</sub>	Load Halfword (64)	RRE	B907	EI
LGR	R <sub>1</sub> ,R <sub>2</sub>	Load (64)	RRE	B904	N
LH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Halfword (32)	RX	48	
LHI	R <sub>1</sub> ,I <sub>2</sub>	Load Halfword Immediate (32)	RI <sub>1</sub>	A78	
LHR	R <sub>1</sub> ,R <sub>2</sub>	Load Halfword (32)	RRE	B927	EI
LHY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Halfword (32)	RXY	E378	LD
LLC	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Logical Character (32)	RXY	E394	EI
LLCR	R <sub>1</sub> ,R <sub>2</sub>	Load Logical Character (32)	RRE	B994	EI
LLGC	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Logical Character	RXY	E390	N
LLGCR	R <sub>1</sub> ,R <sub>2</sub>	Load Logical Character (64)	RRE	B984	EI
LLGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Logical (64<32)	RXY	E316	N
LLGFR	R <sub>1</sub> ,R <sub>2</sub>	Load Logical (64<32)	RRE	B916	N
LLGH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Logical Halfword	RXY	E391	N
LLGHR	R <sub>1</sub> ,R <sub>2</sub>	Load Logical Halfword (64)	RRE	B985	EI
LLGT	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Logical Thirty One Bits	RXY	E317	N
LLGTR	R <sub>1</sub> ,R <sub>2</sub>	Load Logical Thirty One Bits	RRE	B917	N
LLH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Logical Halfword (32)	RXY	E395	EI

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
LLHR	R <sub>1</sub> ,R <sub>2</sub>	Load Logical Halfword (32)	RRE	B995	EI
LLIFH	R <sub>1</sub> ,I <sub>2</sub>	Load Logical Immediate (high)	RIL	C0E	EI
LLIH	R <sub>1</sub> ,I <sub>2</sub>	Load Logical Immediate (high high)	RI <sub>1</sub>	A5C	N
LLILH	R <sub>1</sub> ,I <sub>2</sub>	Load Logical Immediate (high low)	RI <sub>1</sub>	A5D	N
LLILF	R <sub>1</sub> ,I <sub>2</sub>	Load Logical Immediate (low)	RIL	C0F	N
LLILH	R <sub>1</sub> ,I <sub>2</sub>	Load Logical Immediate (low high)	RI <sub>1</sub>	A5E	N
LLILL	R <sub>1</sub> ,I <sub>2</sub>	Load Logical Immediate (low low)	RI <sub>1</sub>	A5F	N
LM	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Multiple (32)	RS <sub>1</sub>	98	
LMD	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> ),D <sub>4</sub> (B <sub>4</sub> )	Load Multiple Disjoint	SS <sub>5</sub>	EF	N
LMG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Multiple (64)	RSY <sub>1</sub>	EB04	N
LMH	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Multiple High	RSY <sub>1</sub>	EB96	N
LMY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Load Multiple (32)	RSY <sub>1</sub>	EB98	LD
LNDBR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (LB)	RRE	B311	c
LNDR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (LH)	RR	21	c
LNEBR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (SB)	RRE	B301	c
LNER	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (SH)	RR	31	c
LNGFR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (64<32)	RRE	B911	c N
LNGR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (64)	RRE	B901	c N
LNR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (32)	RR	11	c
LNXBR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (EB)	RRE	B341	c
LNXR	R <sub>1</sub> ,R <sub>2</sub>	Load Negative (EH)	RRE	B361	c
LPDBR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (LB)	RRE	B310	c
LPDR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (LH)	RR	20	c
LPEBR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (SB)	RRE	B300	c
LPER	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (SH)	RR	30	c
LPGFR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (64<32)	RRE	B910	c N
LPGR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (64)	RRE	B900	c N
LPQ	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Pair from Quadword	RXY	E38F	N
LPR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (32)	RR	10	c
LPSW	D <sub>2</sub> (B <sub>2</sub> )	Load PSW	S	82	pn
LPSWE	D <sub>2</sub> (B <sub>2</sub> )	Load PSW Extended	S	B2B2	pn N
LPTEA	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub> ,M <sub>4</sub>	Load Page-Tagle-Entry Address	RRF <sub>3</sub>	B9AA	c D2
LPXBR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (EB)	RRE	B340	c
LPXR	R <sub>1</sub> ,R <sub>2</sub>	Load Positive (EH)	RRE	B360	c
LR	R <sub>1</sub> ,R <sub>2</sub>	Load (32)	RR	18	
LRA	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Real Address (32)	RX	B1	pc
LRAG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Real Address (64)	RXY	E303	pc N
LRAY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Real Address (32)	RXY	E313	pc LD
LRDR	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (LH<EH)	RR	25	
LRER	R <sub>1</sub> ,R <sub>2</sub>	Load Rounded (SH<LH)	RR	35	
LRV	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Reversed (32)	RXY	E31E	N3
LRVG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Reversed (64)	RXY	E30F	N
LRVGR	R <sub>1</sub> ,R <sub>2</sub>	Load Reversed (64)	RRE	B90F	N
LRVH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Reversed (16)	RXY	E31F	N3
LRVR	R <sub>1</sub> ,R <sub>2</sub>	Load Reversed (32)	RRE	B91F	N3
LT	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load and Test (32)	RXY	E312	c EI
LTDBR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (LB)	RRE	B312	c
LTDR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (LH)	RR	22	c
LTEBR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (SB)	RRE	B302	c
LTER	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (SH)	RR	32	c
LTG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load and Test (64)	RXY	E302	c EI
LTGFR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (64<32)	RRE	B912	c N
LTGR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (64)	RRE	B902	c N

Mne- monic	Operands	Name	For- mat	Op- code	Class & Notes
LTR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (32)	RR	12	c
LTXBR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (EB)	RRE	B342	c
LTXR	R <sub>1</sub> ,R <sub>2</sub>	Load and Test (EH)	RRE	B362	c
LURA	R <sub>1</sub> ,R <sub>2</sub>	Load Using Real Address (32)	RRE	B24B	p
LURAG	R <sub>1</sub> ,R <sub>2</sub>	Load Using Real Address (64)	RRE	B905	p N
LXD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Lengthened (EH<LH)	RXE	ED25	
LXDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Lengthened (EB<LB)	RXE	ED05	
LXDBR	R <sub>1</sub> ,R <sub>2</sub>	Load Lengthened (EB<LB)	RRE	B305	
LXDR	R <sub>1</sub> ,R <sub>2</sub>	Load Lengthened (EH<LH)	RRE	B325	
LXE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Lengthened (EH<SH)	RXE	ED26	
LXEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load Lengthened (EB<SB)	RXE	ED06	
LXEBC	R <sub>1</sub> ,R <sub>2</sub>	Load Lengthened (EB<SB)	RRE	B306	
LXER	R <sub>1</sub> ,R <sub>2</sub>	Load Lengthened (EH<SH)	RRE	B326	
LXR	R <sub>1</sub> ,R <sub>2</sub>	Load (E)	RRE	B365	
LY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Load (32)	RXY	E358	LD
LZDR	R <sub>1</sub>	Load Zero (L)	RRE	B375	
LZER	R <sub>1</sub>	Load Zero (S)	RRE	B374	
LZXR	R <sub>1</sub>	Load Zero (E)	RRE	B376	
M	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (64<32)	RX	5C	
MAD	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Add (LH)	RXF	ED3E	HM
MADB	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Add (LB)	RXF	ED1E	
MADB	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Add (LB)	RRF <sub>1</sub>	B31E	
MADR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Add (LH)	RRF <sub>1</sub>	B33E	HM
MAE	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Add (SH)	RXF	ED2E	HM
MAEB	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Add (SB)	RXF	ED0E	
MAEB	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Add (SB)	RRF <sub>1</sub>	B30E	
MAER	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Add (SH)	RRF <sub>1</sub>	B32E	HM
MAY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Add Unnormalized (EH<LH)	RXF	ED3A	UE
MAYH	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Add Unnormalized (EH <sub>H</sub> <LH)	RXF	ED3C	UE
MAYHR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Add Unnormalized (EH <sub>H</sub> <LH)	RRF <sub>1</sub>	B33C	UE
MAYL	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Add Unnormalized (EH <sub>L</sub> <LH)	RXF	ED38	UE
MAYLR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Add Unnormalized (EH <sub>L</sub> <LH)	RRF <sub>1</sub>	B338	UE
MAYR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Add Unnormalized (EH<LH)	RRF <sub>1</sub>	B33A	UE
MC	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Monitor Call	SI	AF	
MD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (LH)	RX	6C	
MDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (LB)	RXE	ED1C	
MDBR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (LB)	RRE	B31C	
MDE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (LH<SH)	RX	7C	
MDEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (LB<SB)	RXE	ED0C	
MDEBR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (LB<SB)	RRE	B30C	
MDER	R <sub>1</sub> ,R <sub>2</sub>	Multiply (LH<SH)	RR	3C	
MDR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (LH)	RR	2C	
ME	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (LH<SH)	RX	7C	
MEE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (SH)	RXE	ED37	
MEEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (SB)	RXE	ED17	
MEEBR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (SB)	RRE	B317	
MEER	R <sub>1</sub> ,R <sub>2</sub>	Multiply (SH)	RRE	B337	
MER	R <sub>1</sub> ,R <sub>2</sub>	Multiply (LH<SH)	RR	3C	
MGH	R <sub>1</sub> ,I <sub>2</sub>	Multiply Halfword Immediate (64)	RI <sub>1</sub>	A7D	N
MH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Halfword (32)	RX	4C	
MHI	R <sub>1</sub> ,I <sub>2</sub>	Multiply Halfword Immediate (32)	RI <sub>1</sub>	A7C	

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
ML	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Logical (64<32)	RXY	E396	N3
MLG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Logical (128<64)	RXY	E386	N
MLGR	R <sub>1</sub> ,R <sub>2</sub>	Multiply Logical (128<64)	RRE	B986	N
MLR	R <sub>1</sub> ,R <sub>2</sub>	Multiply Logical (64<32)	RRE	B996	N3
MP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Multiply Decimal	SS <sub>2</sub>	FC	
MR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (64<32)	RR	1C	
MS	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Single (32)	RX	71	
MSCH	D <sub>2</sub> (B <sub>2</sub> )	Modify Subchannel	S	B232	pc
MSD	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Subtract (LH)	RXF	ED3F	HM
MSDB	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Subtract (LB)	RXF	ED1F	
MSDBR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Subtract (LB)	RRF <sub>1</sub>	B31F	
MSDR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Subtract (LH)	RRF <sub>1</sub>	B33F	HM
MSE	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Subtract (SH)	RXF	ED2F	HM
MSEB	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply and Subtract (SB)	RXF	ED0F	
MSEBR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Subtract (SB)	RRF <sub>1</sub>	B30F	
MSER	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply and Subtract (SH)	RRF <sub>1</sub>	B32F	HM
MSG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Single (64)	RXY	E30C	N
MSGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Single (64<32)	RXY	E31C	N
MSGFR	R <sub>1</sub> ,R <sub>2</sub>	Multiply Single (64<32)	RRE	B91C	N
MSGR	R <sub>1</sub> ,R <sub>2</sub>	Multiply Single (64)	RRE	B90C	N
MSR	R <sub>1</sub> ,R <sub>2</sub>	Multiply Single (32)	RRE	B252	
MSTA	R <sub>1</sub>	Modify Stacked State	RRE	B247	
MSY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Single (32)	RXY	E351	LD
MVC	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Move (character)	SS <sub>1</sub>	D2	
MVCDK	D <sub>1</sub> (B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Move with Destination key	SSE	E50F	q
MVCIN	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Move Inverse	SS <sub>1</sub>	E8	
MVCK	D <sub>1</sub> (R <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> ),R <sub>3</sub>	Move with Key	SS <sub>4</sub>	D9	qc
MVCL	R <sub>1</sub> ,R <sub>2</sub>	Move Long	RR	0E	ic
MVCLE	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Move Long Extended	RS <sub>1</sub>	A8	c
MVCLU	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Move Long Unicode	RSY <sub>1</sub>	EB8E	c E2
MVCP	D <sub>1</sub> (R <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> ),R <sub>3</sub>	Move to Primary	SS <sub>4</sub>	DA	qc
MVCS	D <sub>1</sub> (R <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> ),R <sub>3</sub>	Move to Secondary	SS <sub>4</sub>	DB	qc
MVCSK	D <sub>1</sub> (B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Move with Source Key	SSE	E50E	q
MVI	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Move (immediate)	SI	92	
MVIY	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Move (immediate)	SIY	EB52	LD
MVN	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Move Numerics	SS <sub>1</sub>	D1	
MVO	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Move with Offset	SS <sub>2</sub>	F1	
MVPG	R <sub>1</sub> ,R <sub>2</sub>	Move Page	RRE	B254	qc
MVST	R <sub>1</sub> ,R <sub>2</sub>	Move String	RRE	B255	c
MVZ	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Move Zones	SS <sub>1</sub>	D3	
MXBR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (EB)	RRE	B34C	
MXD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (EH<LH)	RX	67	
MXDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply (EB<LB)	RXE	ED07	
MXDBR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (EB<LB)	RRE	B307	
MXDR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (EH<LH)	RR	27	
MXR	R <sub>1</sub> ,R <sub>2</sub>	Multiply (EH)	RR	26	
MY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Unnormalized (EH<LH)	RXF	ED3B	UE
MYH	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Unnormalized (EH <sub>H</sub> <LH)	RXF	ED3D	UE
MYHR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply Unnormalized (EH <sub>H</sub> <LH)	RRF <sub>1</sub>	B33D	UE
MYL	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Multiply Unnormalized (EH <sub>L</sub> <LH)	RXF	ED39	UE
MYLR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply Unnormalized (EH <sub>L</sub> <LH)	RRF <sub>1</sub>	B339	UE
MYR	R <sub>1</sub> ,R <sub>3</sub> ,R <sub>2</sub>	Multiply Unnormalized (EH<LH)	RRF <sub>1</sub>	B33B	UE

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
N	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	And (32)	RX	54	c
NC	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	And (character)	SS <sub>1</sub>	D4	c
NG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	And (64)	RXY	E380	c N
NGR	R <sub>1</sub> ,R <sub>2</sub>	And (64)	RRE	B980	c N
NI	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	And (immediate)	SI	94	c
NIHF	R <sub>1</sub> ,I <sub>2</sub>	And Immediate (high)	RIL	C0A	c EI
NIHH	R <sub>1</sub> ,I <sub>2</sub>	And Immediate (high high)	RI <sub>1</sub>	A54	c N
NIHL	R <sub>1</sub> ,I <sub>2</sub>	And Immediate (high low)	RI <sub>1</sub>	A55	c N
NILF	R <sub>1</sub> ,I <sub>2</sub>	And Immediate (low)	RIL	C0B	c EI
NILH	R <sub>1</sub> ,I <sub>2</sub>	And Immediate (low high)	RI <sub>1</sub>	A56	c N
NILL	R <sub>1</sub> ,I <sub>2</sub>	And Immediate (low low)	RI <sub>1</sub>	A57	c N
NIY	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	And (immediate)	SIY	EB54	c LD
NR	R <sub>1</sub> ,R <sub>2</sub>	And (32)	RR	14	c
NY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	And (32)	RXY	E354	c LD
O	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Or (32)	RX	56	c
OC	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Or (character)	SS <sub>1</sub>	D6	c
OG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Or (64)	RXY	E381	c N
OGR	R <sub>1</sub> ,R <sub>2</sub>	Or (64)	RRE	B981	c N
OI	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Or (immediate)	SI	96	c
OIFH	R <sub>1</sub> ,I <sub>2</sub>	Or Immediate (high)	RIL	C0C	c EI
OIHH	R <sub>1</sub> ,I <sub>2</sub>	Or Immediate (high high)	RI <sub>1</sub>	A58	c N
OIHL	R <sub>1</sub> ,I <sub>2</sub>	Or Immediate (high low)	RI <sub>1</sub>	A59	c N
OILF	R <sub>1</sub> ,I <sub>2</sub>	Or Immediate (low)	RIL	C0D	c N
OILH	R <sub>1</sub> ,I <sub>2</sub>	Or Immediate (low high)	RI <sub>1</sub>	A5A	c N
OILL	R <sub>1</sub> ,I <sub>2</sub>	Or Immediate (low low)	RI <sub>1</sub>	A5B	c N
OIY	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Or (immediate)	SIY	EB56	c LD
OR	R <sub>1</sub> R <sub>2</sub>	Or (32)	RR	16	c
OY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Or (32)	RXY	E356	c LD
PACK	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Pack	SS <sub>2</sub>	F2	
PALB		Purge ALB	RRE	B248	p
PC	D <sub>2</sub> (B <sub>2</sub> )	Program Call	S	B218	q
PGIN	R <sub>1</sub> ,R <sub>2</sub>	Page In	RRE	B22E	pc ES
PGOUT	R <sub>1</sub> ,R <sub>2</sub>	Page Out	RRE	B22F	pc ES
PKA	D <sub>1</sub> (B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Pack ASCII	SS <sub>1</sub>	E9	E2
PKU	D <sub>1</sub> (B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Pack Unicode	SS <sub>1</sub>	E1	E2
PLO	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> ),R <sub>3</sub> ,D <sub>4</sub> (B <sub>4</sub> )	Perform Locked Operation	SS <sub>5</sub>	EE	c
PR		Program Return	E	0101	qn
PT	R <sub>1</sub> ,R <sub>2</sub>	Program Transfer	RRE	B228	q
PTFF		Perform Timing-Facility Function	E	0104	qc
PTI	R <sub>1</sub> ,R <sub>2</sub>	Program Transfer with Instance	RRE	B99E	q RA
PTLB		Purge TLB	S	B20D	p
RCHP		Reset Channel Path	S	B23B	pc
RLL	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Rotate Left Single Logical (32)	RSY <sub>1</sub>	EB1D	N3
RLLG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Rotate Left Single Logical (64)	RSY <sub>1</sub>	EB1C	N
RP	D <sub>2</sub> (B <sub>2</sub> )	Resume Program	S	B277	qn
RRBE	R <sub>1</sub> ,R <sub>2</sub>	Reset Reference Bit Extended	RRE	B22A	pc
RSCH		Resume Subchannel	S	B238	pc
S	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract (32)	RX	5B	c
SAC	D <sub>2</sub> (B <sub>2</sub> )	Set Address Space Control	S	B219	q
SACF	D <sub>2</sub> (B <sub>2</sub> )	Set Address Space Control Fast	S	B279	q
SAL		Set Address Limit	S	B237	p
SAM24		Set Addressing Mode (24)	E	010C	N3
SAM31		Set Addressing Mode (31)	E	010D	N3
SAM64		Set Addressing Mode (64)	E	010E	N
SAR	R <sub>1</sub> ,R <sub>2</sub>	Set Access	RRE	B24E	
SCHM		Set Channel Monitor	S	B23C	p

Mne-monic	Operands	Name	For-mat	Op-code	Class & Notes
SCK	D <sub>2</sub> (B <sub>2</sub> )	Set Clock	S	B204	pc
SCKC	D <sub>2</sub> (B <sub>2</sub> )	Set Clock Comparator	S	B206	p
SCKPF		Set Clock Programmable Field	E	0107	p
SD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Normalized (LH)	RX	6B	c
SDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract (LB)	RXE	ED1B	c
SDBR	R <sub>1</sub> ,R <sub>2</sub>	Subtract (LB)	RRE	B31B	c
SDR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Normalized (LH)	RR	2B	c
SE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Normalized (SH)	RX	7B	c
SEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract (SB)	RXE	ED0B	c
SEBR	R <sub>1</sub> ,R <sub>2</sub>	Subtract (SB)	RRE	B30B	c
SER	R <sub>1</sub> ,R <sub>2</sub>	Subtract Normalized (SH)	RR	3B	c
SFPC	R <sub>1</sub>	Set FPC	RRE	B384	
SG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract (64)	RXY	E309	c N
SGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract (64<32)	RXY	E319	c N
SGFR	R <sub>1</sub> ,R <sub>2</sub>	Subtract (64<32)	RRE	B919	c N
SGR	R <sub>1</sub> ,R <sub>2</sub>	Subtract (64)	RRE	B909	c N
SH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Halfword	RX	4B	c
SHY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Halfword	RXY	E37B	c LD
SIE	D <sub>2</sub> (B <sub>2</sub> )	Start Interpretive Execution	S	B214	ip
SIGP	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Signal Processor	RS <sub>1</sub>	AE	pc
SL	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Logical (32)	RX	5F	c
SLA	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Left Single (32)	RS <sub>1</sub>	8B	c
SLAG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Left Single (64)	RSY <sub>1</sub>	EB0B	c N
SLB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Logical with Borrow (32)	RXY	E399	c N3
SLBG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Logical with Borrow (64)	RXY	E389	c N
SLBGR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Logical with Borrow (64)	RRE	B989	c N
SLBR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Logical with Borrow (32)	RRE	B999	c N3
SLDA	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Left Double	RS <sub>1</sub>	8F	c
SLDL	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Left Double Logical	RS <sub>1</sub>	8D	
SLFI	R <sub>1</sub> ,I <sub>2</sub>	Subtract Logical Immediate (32)	RIL	C25	c EI
SLG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Logical (64)	RXY	E30B	c N
SLGF	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Logical (64<32)	RXY	E31B	c N
SLGFI	R <sub>1</sub> ,I <sub>2</sub>	Subtract Logical Immediate (64<32)	RIL	C24	c EI
SLGFR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Logical (64<32)	RRE	B91B	c N
SLGR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Logical (64)	RRE	B90B	c N
SLL	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Left Single Logical (32)	RS <sub>1</sub>	89	
SLLG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Left Single Logical (64)	RSY <sub>1</sub>	EB0D	N
SLR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Logical (32)	RR	1F	c
SLY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Logical (32)	RXY	E35F	c LD
SP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Subtract Decimal	SS <sub>2</sub>	FB	c
SPKA	D <sub>2</sub> (B <sub>2</sub> )	Set PSW Key from Address	S	B20A	q
SPM	R <sub>1</sub>	Set Program Mask	RR	04	n
SPT	D <sub>2</sub> (B <sub>2</sub> )	Set CPU Timer	S	B208	p
SPX	D <sub>2</sub> (B <sub>2</sub> )	Set Prefix	S	B210	p
SQD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Square Root (LH)	RXE	ED35	
SQDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Square Root (LB)	RXE	ED15	
SQDBR	R <sub>1</sub> ,R <sub>2</sub>	Square Root (LB)	RRE	B315	
SQDR	R <sub>1</sub> ,R <sub>2</sub>	Square Root (LH)	RRE	B244	
SQE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Square Root (SH)	RXE	ED34	
SQEBS	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Square Root (SB)	RXE	ED14	
SQEBR	R <sub>1</sub> ,R <sub>2</sub>	Square Root (SB)	RRE	B314	
SQER	R <sub>1</sub> ,R <sub>2</sub>	Square Root (SH)	RRE	B245	

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
SQXBR	R <sub>1</sub> ,R <sub>2</sub>	Square Root (EB)	RRE	B316	
SQXR	R <sub>1</sub> ,R <sub>2</sub>	Square Root (EH)	RRE	B336	
SR	R <sub>1</sub> ,R <sub>2</sub>	Subtract (32)	RR	1B	c
SRA	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Right Single (32)	RS <sub>1</sub>	8A	c
SRAG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Right Single (64)	RSY <sub>1</sub>	EB0A	c N
SRDA	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Right Double	RS <sub>1</sub>	8E	c
SRDL	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Right Double Logical	RS <sub>1</sub>	8C	
SRL	R <sub>1</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Right Single Logical (32)	RS <sub>1</sub>	88	
SRLG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Shift Right Single Logical (64)	RSY <sub>1</sub>	EB0C	N
SRNM	D <sub>2</sub> (B <sub>2</sub> )	Set Rounding Mode	S	B299	
SRP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> ),I <sub>3</sub>	Shift and Round Decimal	SS <sub>3</sub>	F0	c
SRST	R <sub>1</sub> ,R <sub>2</sub>	Search String	RRE	B25E	c
SRSTU	R <sub>1</sub> ,R <sub>2</sub>	Search String Unicode	RRE	B2BE	c E3
SSAIR	R <sub>1</sub>	Set Secondary ASN with Instance	RRE	B99F	RA
SSAR	R <sub>1</sub>	Set Secondary ASN	RRE	B225	
SSCH	D <sub>2</sub> (B <sub>2</sub> )	Start Subchannel	S	B233	pc
SSKE	R <sub>1</sub> ,R <sub>2</sub>	Set Storage Key Extended	RRE	B22B	p
SSM	D <sub>2</sub> (B <sub>2</sub> )	Set System Mask	S	80	p
ST	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store (32)	RX	50	
STAM	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Access Multiple	RS <sub>1</sub>	9B	
STAMY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Access Multiple	RSY <sub>1</sub>	EB9B	LD
STAP	D <sub>2</sub> (B <sub>2</sub> )	Store CPU Address	S	B212	p
STC	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Character	RX	42	
STCK	D <sub>2</sub> (B <sub>2</sub> )	Store Clock	S	B205	c
STCKC	D <sub>2</sub> (B <sub>2</sub> )	Store Clock Comparator	S	B207	p
STCKE	D <sub>2</sub> (B <sub>2</sub> )	Store Clock Extended	S	B278	c
STCKF	D <sub>2</sub> (B <sub>2</sub> )	Store Clock Fast	S	B27C	c SC
STCM	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Characters under Mask (low)	RS <sub>2</sub>	BE	
STCMH	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Characters under Mask (high)	RSY <sub>1</sub>	EB2C	N
STCMY	R <sub>1</sub> ,M <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Characters under Mask (low)	RSY <sub>2</sub>	EB2D	LD
STCP	D <sub>2</sub> (B <sub>2</sub> )	Store Channel Path Status	S	B23A	p
STCRW	D <sub>2</sub> (B <sub>2</sub> )	Store Channel Report Word	S	B239	pc
STCTG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Control (64)	RSY <sub>1</sub>	EB25	p N
STCTL	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Control (32)	RS <sub>1</sub>	B6	p
STCY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Character	RXY	E372	LD
STD	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store (L)	RX	60	
STDY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store (L)	RXY	ED67	LD
STE	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store (S)	RX	70	
STEY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store (S)	RXY	ED66	LD
STFL	D <sub>2</sub> (B <sub>2</sub> )	Store Facility List	S	B2B1	p N3
STFLE	D <sub>2</sub> (B <sub>2</sub> )	Store Facility List Extended	S	B2B0	c FL
STFPC	D <sub>2</sub> (B <sub>2</sub> )	Store FPC	S	B29C	
STG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store (64)	RXY	E324	N
STH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Halfword	RX	40	
STHY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Halfword	RXY	E370	LD
STIDP	D <sub>2</sub> (B <sub>2</sub> )	Store CPU ID	S	B202	p
STM	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Multiple (32)	RS <sub>1</sub>	90	
STMG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Multiple (64)	RSY <sub>1</sub>	EB24	N
STMH	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Multiple High	RSY <sub>1</sub>	EB26	N
STMY	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Store Multiple (32)	RSY <sub>1</sub>	EB90	LD
STNSM	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Store Then And System Mask	SI	AC	p
STOSM	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Store Then Or System Mask	SI	AD	p
STPQ	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Pair to Quadword	RXY	E38E	N
STPT	D <sub>2</sub> (B <sub>2</sub> )	Store CPU Timer	S	B209	p

Mnemonic	Operands	Name	Format	Op-code	Class & Notes
STPX	D <sub>2</sub> (B <sub>2</sub> )	Store Prefix	S	B211	p
STRAG	D <sub>1</sub> (B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Store Real Address	SSE	E502	p N
STRV	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Reversed (32)	RXY	E33E	N3
STRVG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Reversed (64)	RXY	E32F	N
STRVH	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store Reversed (16)	RXY	E33F	N3
STSCH	D <sub>2</sub> (B <sub>2</sub> )	Store Subchannel	S	B234	pc
STSI	D <sub>2</sub> (B <sub>2</sub> )	Store System Information	S	B27D	pc
STURA	R <sub>1</sub> ,R <sub>2</sub>	Store Using Real Address (32)	RRE	B246	p
STURG	R <sub>1</sub> ,R <sub>2</sub>	Store Using Real Address (64)	RRE	B925	p N
STY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Store (32)	RXY	E350	LD
SU	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Unnormalized (SH)	RX	7F	c
SUR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Unnormalized (SH)	RR	3F	c
SVC	I	Supervisor Call	I	0A	
SW	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract Unnormalized (LH)	RX	6F	c
SWR	R <sub>1</sub> ,R <sub>2</sub>	Subtract Unnormalized (LH)	RR	2F	c
SXBR	R <sub>1</sub> ,D <sub>2</sub>	Subtract (EB)	RRE	B34B	c
SXR	R <sub>1</sub> ,D <sub>2</sub>	Subtract Normalized (EH)	RR	37	c
SY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Subtract (32)	RXY	E35B	c LD
TAM		Test Addressing Mode	E	010B	c N3
TAR	R <sub>1</sub> ,R <sub>2</sub>	Test Access	RRE	B24C	c
TB	R <sub>1</sub> ,R <sub>2</sub>	Test Block	RRE	B22C	ipc
TBDR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert HFP to BFP (LB<LH)	RRF <sub>2</sub>	B351	c
TBEDR	R <sub>1</sub> ,M <sub>3</sub> ,R <sub>2</sub>	Convert HFP to BFP (SB<LH)	RRF <sub>2</sub>	B350	c
TCDB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Test Data Class (LB)	RXE	ED11	c
TCEB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Test Data Class (SB)	RXE	ED10	c
TCXB	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Test Data Class (EB)	RXE	ED12	c
THDER	R <sub>1</sub> ,R <sub>2</sub>	Convert BFP to HFP (LH<SB)	RRE	B358	c
THDR	R <sub>1</sub> ,R <sub>2</sub>	Convert BFP to HFP (LH<LB)	RRE	B359	c
TM	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Test under Mask	SI	91	c
TMH	R <sub>1</sub> ,I <sub>2</sub>	Test under Mask High	RI <sub>1</sub>	A70	c
TMHH	R <sub>1</sub> ,I <sub>2</sub>	Test under Mask (high high)	RI <sub>1</sub>	A72	c N
TMHL	R <sub>1</sub> ,I <sub>2</sub>	Test under Mask (high low)	RI <sub>1</sub>	A73	c N
TML	R <sub>1</sub> ,I <sub>2</sub>	Test under Mask Low	RI <sub>1</sub>	A71	c
TMLH	R <sub>1</sub> ,I <sub>2</sub>	Test under Mask (low high)	RI <sub>1</sub>	A70	c N
TMLL	R <sub>1</sub> ,I <sub>2</sub>	Test under Mask (low low)	RI <sub>1</sub>	A71	c N
TMY	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Test under Mask	SIY	EB51	c LD
TP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> )	Test Decimal	RSL	EBC0	c E2
TPI	D <sub>2</sub> (B <sub>2</sub> )	Test Pending Interruption	S	B236	pc
TPROT	D <sub>1</sub> (B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Test Protection	SSE	E501	pc
TR	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Translate	SS <sub>1</sub>	DC	
TRACE	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Trace (32)	RS <sub>1</sub>	99	p
TRACG	R <sub>1</sub> ,R <sub>3</sub> ,D <sub>2</sub> (B <sub>2</sub> )	Trace (64)	RSY <sub>1</sub>	EB0F	p N
TRAP2		Trap	E	01FF	
TRAP4	D <sub>2</sub> (B <sub>2</sub> )	Trap	S	B2FF	
TRE	R <sub>1</sub> ,R <sub>2</sub>	Translate Extended	RRE	B2A5	c
TROO	R <sub>1</sub> ,R <sub>2</sub> [,M <sub>3</sub> ]	Translate One to One	RRF <sub>2</sub>	B993	c E2
TROT	R <sub>1</sub> ,R <sub>2</sub> [,M <sub>3</sub> ]	Translate One to Two	RRF <sub>2</sub>	B992	c E2
TRT	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Translate and Test	SS <sub>1</sub>	DD	c
TRTO	R <sub>1</sub> ,R <sub>2</sub> [,M <sub>3</sub> ]	Translate Two to One	RRF <sub>2</sub>	B991	c E2
TRTR	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Translate and Test Reverse	SS <sub>1</sub>	D0	c E3
TRTT	R <sub>1</sub> ,R <sub>2</sub> [,M <sub>3</sub> ]	Translate Two to Two	RRF <sub>2</sub>	B990	c E2
TS	D <sub>2</sub> (B <sub>2</sub> )	Test and Set	S	93	c
TSCH	D <sub>2</sub> (B <sub>2</sub> )	Test Subchannel	S	B235	pc
UNPK	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Unpack	SS <sub>2</sub>	F3	
UNPKA	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Unpack ASCII	SS <sub>1</sub>	EA	c E2
UNPKU	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Unpack Unicode	SS <sub>1</sub>	E2	c E2
UPT		Update Tree	E	0102	ic

Mne- monic	Operands	Name	For- mat	Op- code	Class & Notes
X	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Exclusive Or (32)	RX	57	c
XC	D <sub>1</sub> (L,B <sub>1</sub> ),D <sub>2</sub> (B <sub>2</sub> )	Exclusive Or (character)	SS <sub>1</sub>	D7	c
XG	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Exclusive Or (64)	RXY	E382	c N
XGR	R <sub>1</sub> ,R <sub>2</sub>	Exclusive Or (64)	RRE	B982	c N
XI	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Exclusive Or (immediate)	SI	97	c
XIY	D <sub>1</sub> (B <sub>1</sub> ),I <sub>2</sub>	Exclusive Or (immediate)	SIY	EB57	c LD
XR	R <sub>1</sub> ,R <sub>2</sub>	Exclusive Or (32)	RR	17	c
XSCH		Cancel Subchannel	S	B276	pc
XY	R <sub>1</sub> ,D <sub>2</sub> (X <sub>2</sub> ,B <sub>2</sub> )	Exclusive Or (32)	RXY	E357	c LD
ZAP	D <sub>1</sub> (L <sub>1</sub> ,B <sub>1</sub> ),D <sub>2</sub> (L <sub>2</sub> ,B <sub>2</sub> )	Zero and Add	SS <sub>2</sub>	F8	c

#### Floating-Point Operand Lengths and Types:

#### Notes:

E	Extended (binary or hex)	c	Condition code set
EB	Extended binary	i	Interruptible instruction
EH	Extended hex	n	New condition code loaded
EH <sub>L</sub>	Extended hex (low-order part)	p	Privileged instruction
EH <sub>H</sub>	Extended hex (high-order part)	q	Semiprivililed instruction
L	Long (binary or hex)	u	Condition code is unpredictable
LB	Long binary	D2	DAT-enhancement facility 2
LH	Long hex	EI	Extended-immediate facility
S	Short (binary or hex)	E2	Extended-translation facility 2
SB	Short binary	E3	Extended-translation facility 3
SH	Short hex	ES	Expanded-storage facility
		FL	Store-facility-list-extended facility
		HM	HFP multiply-and-add/subtract facility
		LD	Long-displacement facility
		N	New in z/Architecture
		MS	Message-security assist
		N3	New in z/Architecture and added to ESA/390
		RA	ASN-and-LX-reuse facility
		SC	Store-clock-fast facility
		UE	HFP unnormalized-extension facility

## Machine Instructions by Operation Code

Op-Code	Mne-monic	Op-Code	Mne-monic	Op-Code	Mne-monic
0101	PR	41	LA	96	OI
0102	UPT	42	STC	97	XI
0104	PTFF	43	IC	98	LM
0107	SCKPF	44	EX	99	TRACE
010B	TAM	45	BAL	9A	LAM
010C	SAM24	46	BCT	9B	STAM
010D	SAM31	47	BC	A50	IIHH
010E	SAM64	48	LH	A51	IIHL
01FF	TRAP2	49	CH	A52	IILH
04	SPM	4A	AH	A53	ILL
05	BALR	4B	SH	A54	NIHH
06	BCTR	4C	MH	A55	NIHL
07	BCR	4D	BAS	A56	NILH
0A	SVC	4E	CVD	A57	NILL
0B	BSM	4F	CVB	A58	OIHH
0C	BASSM	50	ST	A59	OIHL
0D	BASR	51	LAE	A5A	OILH
0E	MVCL	54	N	A5B	OILL
0F	CLCL	55	CL	A5C	LLIHH
10	LPR	56	O	A5D	LLIHL
11	LNR	57	X	A5E	LLILH
12	LTR	58	L	A5F	LLILL
13	LCR	59	C	A70	TMLH
14	NR	5A	A	A70	TMH
15	CLR	5B	S	A71	TMLL
16	OR	5C	M	A71	TML
17	XR	5D	D	A72	TMHH
18	LR	5E	AL	A73	TMHL
19	CR	5F	SL	A74	BRC
1A	AR	60	STD	A75	BRAS
1B	SR	67	MXD	A76	BRCT
1C	MR	68	LD	A77	BRCTG
1D	DR	69	CD	A78	LHI
1E	ALR	6A	AD	A79	LGHI
1F	SLR	6B	SD	A7A	AHI
20	LPDR	6C	MD	A7B	AGHI
21	LNDR	6D	DD	A7C	MHI
22	LTDR	6E	AW	A7D	MGHI
23	LCDR	6F	SW	A7E	CHI
24	HDR	70	STE	A7F	CGHI
25	LDXR	71	MS	A8	MVCLE
25	LRDR	78	LE	A9	CLCLE
26	MXR	79	CE	AC	STNSM
27	MXDR	7A	AE	AD	STOSM
28	LDR	7B	SE	AE	SIGP
29	CDR	7C	MDE	AF	MC
2A	ADR	7C	ME	B1	LRA
2B	SDR	7D	DE	B202	STIDP
2C	MDR	7E	AU	B204	SCK
2D	DDR	7F	SU	B205	STCK
2E	AWR	80	SSM	B206	SCKC
2F	SWR	82	LPSW	B207	STCKC
30	LPER	83	Diagnose	B208	SPT
31	LNER	84	BRXH	B209	STPT
32	LTER	85	BRXLE	B20A	SPKA
33	LCER	86	BXH	B20B	IPK
34	HER	87	BXLE	B20D	PTLB
35	LEDR	88	SRL	B210	SPX
35	LRER	89	SLL	B211	STPX
36	AXR	8A	SRA	B212	STAP
37	SXR	8B	SLA	B214	SIE
38	LER	8C	SRDL	B218	PC
39	CER	8D	SLDL	B219	SAC
3A	AER	8E	SRDA	B21A	CFC
3B	SER	8F	SLDA	B221	IPTE
3C	MDER	90	STM	B222	IPM
3C	MER	91	TM	B223	IVSK
3D	DER	92	MVI	B224	IAC
3E	AUR	93	TS	B225	SSAR
3F	SUR	94	NI	B226	EPAR
40	STH	95	CLI	B227	ESAR

Op Code	Mne-monic	Op Code	Mne-monic	Op Code	Mne-monic
B228	PT	B30C	MDEBR	B398	CFEBR
B229	ISKE	B30D	DEBR	B399	CFDBR
B22A	RRBE	B30E	MAEBR	B39A	CFXBR
B22B	SSKE	B30F	MSEBR	B3A4	CEGBR
B22C	TB	B310	LPDBR	B3A5	CDGBR
B22D	DXR	B311	LNDBR	B3A6	CXGBR
B22E	PGIN	B312	LTDBR	B3A8	CGEBR
B22F	PGOUT	B313	LCDBR	B3A9	CGDBR
B230	CSCH	B314	SQEBR	B3AA	CGXBR
B231	HSCH	B315	SQDBR	B3B4	CEFR
B232	MSCH	B316	SQXBR	B3B5	CDFR
B233	SSCH	B317	MEEBR	B3B6	CXFR
B234	STSCH	B318	KDBR	B3B8	CFER
B235	TSCH	B319	CDBR	B3B9	CFDR
B236	TPI	B31A	ADBR	B3BA	CFXR
B237	SAL	B31B	SDBR	B3C4	CEGR
B238	RSCH	B31C	MDBR	B3C5	CDGR
B239	STCRW	B31D	DDBR	B3C6	CXGR
B23A	STCPS	B31E	MADBR	B3C8	CGER
B23B	RCHP	B31F	MSDBR	B3C9	CGDR
B23C	SCHM	B324	LDER	B3CA	CGXR
B240	BAKR	B325	LXDR	B6	STCTL
B241	CKSM	B326	LXER	B7	LCTL
B244	SQDR	B32E	MAER	B900	LPGR
B245	SQER	B32F	MSER	B901	LNGR
B246	STURA	B336	SQXR	B902	LTGR
B247	MSTA	B337	MEER	B903	LCGR
B248	PALB	B338	MAYLR	B904	LGR
B249	EREG	B339	MYLR	B905	LURAG
B24A	ESTA	B33A	MAYR	B906	LGBR
B24B	LURA	B33B	MYR	B907	LGHR
B24C	TAR	B33C	MAYHR	B908	AGR
B24D	CPYA	B33D	MYHR	B909	SGR
B24E	SAR	B33E	MADR	B90A	ALGR
B24F	EAR	B33F	MSDR	B90B	SLGR
B250	CSP	B340	LPXBR	B90C	MSGR
B252	MSR	B341	LNXBR	B90D	DSGR
B254	MVPG	B342	LTXBR	B90E	EREGG
B255	MVST	B343	LCXBR	B90F	LRVGR
B257	CUSE	B344	LEDBR	B910	LPGFR
B258	BSG	B345	LDXBR	B911	LNGFR
B25A	BSA	B346	LEXBR	B912	LTGFR
B25D	CLST	B347	FIXBR	B913	LCGFR
B25E	SRST	B348	KXBR	B914	LGFR
B263	CMPSC	B349	CXBR	B916	LLGFR
B276	XSCH	B34A	AXB	B917	LLGTR
B277	RP	B34B	SXBR	B918	AGFR
B278	STCKE	B34C	MXBR	B919	SGFR
B279	SACF	B34D	DXBR	B91A	ALGFR
B27C	STCKF	B350	TBEDR	B91B	SLGFR
B27D	STSI	B351	TBDR	B91C	MSGFR
B299	SRNM	B353	DIEBR	B91D	DSGFR
B29C	STFPC	B357	FIEBR	B91E	KMAC
B29D	LFPC	B358	THDER	B91F	LRVR
B2A5	TRE	B359	THDR	B920	CGR
B2A6	CU21	B35B	DIDBR	B921	CLGR
B2A6	CUUTF	B35F	FIDBR	B925	STURG
B2A7	CU12	B360	LPXR	B926	LBR
B2A7	CUTFU	B361	LNXR	B927	LHR
B2B0	STFLE	B362	LTXR	B92E	KM
B2B1	STFL	B363	LCXR	B92F	KMC
B2B2	LPSWE	B365	LXR	B930	CGFR
B2FF	TRAP4	B366	LEXR	B931	CLGFR
B300	LPEBR	B367	FIXR	B93E	KIMD
B301	LNEBR	B369	CXR	B93F	KLMD
B302	LTEBR	B374	LZER	B946	BCTGR
B303	LCEBR	B375	LZDR	B980	NGR
B304	LDEBR	B376	LZXR	B981	OGR
B305	LXDBR	B377	FIER	B982	XGR
B306	LXEBR	B37F	FIDR	B983	FLOGR
B307	MXDBR	B384	SFPC	B984	LLGCR
B308	KEBR	B38C	EFPC	B985	LLGHR
B309	CEBR	B394	CEFBR	B986	MLGR
B30A	AEBR	B395	CDFBR	B987	DLGR
B30B	SEBR	B396	CXFBR	B988	ALCGR

Op-Code	Mne-monic	Op-Code	Mne-monic	Op-Code	Mne-monic
B989	SLBGR	E308	AG	E50E	MVCSK
B98A	CSPG	E309	SG	E50F	MVCBK
B98D	EPSW	E30A	ALG	E8	MVCIN
B98E	IDTE	E30B	SLG	E9	PKA
B990	TRTT	E30C	MSG	EA	UNPKA
B991	TRTO	E30D	DSG	EB04	LMG
B992	TROT	E30E	CVBG	EB0A	SRAG
B993	TROO	E30F	LRVG	EB0B	SLAG
B994	LLCR	E312	LT	EB0C	SRLG
B995	LLHR	E313	LRAY	EB0D	SLLG
B996	MLR	E314	LGF	EB0F	TRACG
B997	DLR	E315	LGH	EB14	CSY
B998	ALCR	E316	LLGF	EB1C	RLLG
B999	SLBR	E317	LLGT	EB1D	RLL
B99A	EPAIR	E318	AGF	EB20	CLMH
B99B	ESAIR	E319	SGF	EB21	CLMY
B99D	ESEA	E31A	ALGF	EB24	STMG
B99E	PTI	E31B	SLGF	EB25	STCTG
B99F	SSAIR	E31C	MSGF	EB26	STMH
B9AA	LPTEA	E31D	DSGF	EB2C	STCMH
B9B0	CU14	E31E	LRV	EB2D	STCMY
B9B1	CU24	E31F	LRVH	EB2F	LCTLG
B9B2	CU41	E320	CG	EB30	CSG
B9B3	CU42	E321	CLG	EB31	CDSY
B9BE	SRSTU	E324	STG	EB3E	CDSG
BA	CS	E326	CVDY	EB44	BXHG
BB	CDS	E32E	CVDG	EB45	BXLEG
BD	CLM	E32F	STRVG	EB51	TMY
BE	STCM	E330	CGF	EB52	MVIY
BF	ICM	E331	CLGF	EB54	NIY
C00	LARL	E33E	STRV	EB55	CLIY
C01	LGFI	E33F	STRVH	EB56	OIY
C04	BRCL	E346	BCTG	EB57	XIY
C05	BRASL	E350	STY	EB80	ICMH
C06	XIHF	E351	MSY	EB81	ICMY
C07	XILF	E354	NY	EB8E	MVCLU
C08	IIHF	E355	CLY	EB8F	CLCLU
C09	IILF	E356	OY	EB90	STMY
COA	NIHF	E357	XY	EB96	LMH
C0B	NILF	E358	LY	EB98	LMY
C0C	OIHF	E359	CY	EB9A	LAMY
C0D	OILF	E35A	AY	EB9B	STAMY
C0E	LLIHF	E35B	SY	EBC0	TP
C0F	LLILF	E35E	ALY	EC44	BRXHG
C24	SLGFI	E35F	SLY	EC45	BRXLG
C25	SLFI	E370	STHY	ED04	LDEB
C28	AGFI	E371	LAY	ED05	LXDB
C29	AFI	E372	STCY	ED06	LXEB
C2A	ALGFI	E373	ICY	ED07	MXDB
C2B	ALFI	E376	LB	ED08	KEB
C2C	CGFI	E377	LGB	ED09	CEB
C2D	CFI	E378	LHY	ED0A	AEB
C2E	CLGFI	E379	CHY	ED0B	SEB
C2F	CLFI	E37A	AHY	ED0C	MDEB
D0	TRTR	E37B	SHY	ED0D	DEB
D1	MVN	E380	NG	ED0E	MAEB
D2	MVC	E381	OG	ED0F	MSEB
D3	MVZ	E382	XG	ED10	TCEB
D4	NC	E386	MLG	ED11	TCDB
D5	CLC	E387	DLG	ED12	TCXB
D6	OC	E388	ALCG	ED14	SQEB
D7	XC	E389	SLBG	ED15	SQDB
D9	MVCK	E38E	STPQ	ED17	MEEB
DA	MVCP	E38F	LPQ	ED18	KDB
DB	MVCS	E390	LLGC	ED19	CDB
DC	TR	E391	LLGH	ED1A	ADB
DD	TRT	E394	LLC	ED1B	SDB
DE	ED	E395	LLH	ED1C	MDB
DF	EDMK	E396	ML	ED1D	DDB
E1	PKU	E397	DL	ED1E	MADB
E2	UNPKU	E398	ALC	ED1F	MSDB
E302	LTG	E399	SLB	ED24	LDE
E303	LRAG	E500	LASP	ED25	LXD
E304	LG	E501	TPROT	ED26	LXE
E306	CVBY	E502	STRAG	ED2E	MAE

Op Code	Mne- monic
ED2F	MSE
ED34	SQE
ED35	SQD
ED37	MEE
ED38	MAYL
ED39	MYL
ED3A	MAY
ED3B	MY
ED3C	MAYH
ED3D	MYH
ED3E	MAD
ED3F	MSD
ED64	LEY
ED65	LDY
ED66	STEY
ED67	STDY
EE	PLO
EF	LMD
F0	SRP
F1	MVO
F2	PACK
F3	UNPK
F8	ZAP
F9	CP
FA	AP
FB	SP
FC	MP
FD	DP

## Condition Codes

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
<b>General Instructions</b>				
Add	Zero	< Zero	> Zero	Overflow
Add Halfword	Zero	< Zero	> Zero	Overflow
Add Halfword Immediate	Zero	< Zero	> Zero	Overflow
Add Logical	Zero, no carry	Not zero, no carry	Zero, carry	Not zero, carry
Add Logical with Carry	Zero, no carry	Not zero, no carry	Zero, carry	Not zero, carry
And	Zero	Not zero	—	—
And Immediate	ANDed bits zero	ANDed bits not zero	—	—
Checksum	Checksum complete	—	—	CPU-determined completion
Cipher Message	Normal completion	—	—	Partial completion
Cipher Message with Chaining	Normal completion	—	—	Partial completion
Compare	Equal	First op low	First op high	—
Compare and Form Codeword	Equal	First op low and ctl = 0, or first op high and ctl = 1	First op high and ctl = 0, or first op low and ctl = 1	—
Compare and Swap	Equal	Not equal	—	—
Compare Double and Swap	Equal	Not equal	—	—
Compare Halfword	Equal	First op low	First op high	—
Compare Halfword Immediate	Equal	First op low	First op high	—
Compare Logical	Equal	First op low	First op high	—
Compare Logical Characters under Mask	Equal, or Mask is zero	First op low	First op high	—
Compare Logical Long	Equal	First op low	First op high	—
Compare Logical Long Extended	Equal	First op low	First op high	CPU-determined completion
Compare Logical Long Unicode	Equal	First op low	First op high	CPU-determined completion
Compare Logical String	Equal	First op low	First op high	CPU-determined completion
Compare until Substring Equal	Equal sub-string	Last bytes equal	Last bytes unequal	CPU-determined completion
Compression Call	Second op end	First op end, not second op end	—	CPU-determined completion
Compute Intermed. Message Digest	Normal completion	—	—	Partial completion
Compute Last Message Digest	Normal completion	—	—	Partial completion
Compute Message Authen. Code	Normal completion	—	—	Partial completion
Convert Unicode to UTF-8	Data processed	First op full	—	CPU-determined completion
Convert UTF-8 to Unicode	Data processed	First op full	—	CPU-determined completion
Exclusive Or	Zero	Not zero	—	—
Exclusive Or Immediate	XORed bits zero	XORed bits not zero	—	—
Find Leftmost One	No one bit found	—	One bit found	—

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Insert Characters under Mask	All zero, or mask is zero	Leftmost bit = 1	Not zero, but with leftmost bit = 0	—
Load and Test	Zero	< Zero	> Zero	—
Load Complement	Zero	< Zero	> Zero	Overflow
Load Negative	Zero	< Zero	—	—
Load Positive	Zero	—	> Zero	Overflow
Load Page-Table-Entry Address	Address returned; STE.P=0	Address returned; STE.P=1	Invalid bit one in RTE or STE.	Exception condition exists.
Move Long	Operand lengths equal	First op shorter	First op longer	Overlap
Move Long Extended	Operand lengths equal	First op shorter	First op longer	CPU-determined completion
Move Long Unicode	Operand lengths equal	First op shorter	First op longer	CPU-determined completion
Move Page	Data moved	First op invalid, both valid in ES, locked, or ES error	Second op invalid	—
Move String	—	Second op moved	—	CPU-determined completion
Or	Zero	Not zero	—	—
Or Immediate	ORed bits zero	ORed bits not zero	—	—
Perform Locked Operation (test bit zero)	Equal	First op not equal	First op equal, third op not equal	—
Perform Locked Operation (test bit one)	Code valid	—	—	Code invalid
Perform Timing-Facility Function	Function performed	—	—	Function not avail.
Search String, Search String Unicode	—	Character found	Character not found	CPU-determined completion
Set Program Mask	See Note	See Note	See Note	See Note
Shift Left Double	Zero	< Zero	> Zero	Overflow
Shift Left Single	Zero	< Zero	> Zero	Overflow
Shift Right Double	Zero	< Zero	> Zero	—
Shift Right Single	Zero	< Zero	> Zero	—
Store Clock (STCK, STCKE or STCKF)	Set state	Not-set state	Error state	Stopped state or not operational
Store Facility List Extended	Complete list stored	—	—	Incomplete list stored
Subtract	Zero	< Zero	> Zero	Overflow
Subtract Halfword	Zero	< Zero	> Zero	Overflow
Subtract Logical	—	Not zero, borrow	Zero, no borrow	Not zero, no borrow
Subtract Logical with Borrow	Zero, borrow	Not zero, borrow	Zero, no borrow	Not zero, no borrow
Test Addressing Mode	24-bit mode	31-bit mode	—	—
Test and Set	Leftmost bit zero	Leftmost bit one	—	—
Test under Mask (TM)	All zeros, or mask is zero	Mixed 0's and 1's	—	All ones
Test under Mask (TMH, TMHH, TMHL, TML, TMLH, TMLL)	All zeros or mask is zero	Mixed 0's and 1's and leftmost bit zero	Mixed 0's and 1's and leftmost bit one	All ones
Test under Mask High, Low	All zeros or mask is zero	Mixed 0's and 1's and leftmost bit zero	Mixed 0's and 1's and leftmost bit one	All ones
Translate and Test, Translate and Test Reverse	All zeros	Not zero, scan incomplete	Not zero, scan complete	—

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Translate Extended	Data processed	First op byte equal test byte	—	CPU-determined completion
Translate One to One, One to Two, Two to One, Two to Two	Character not found	Character found	—	CPU determined completion
Unpack ASCII	Sign plus	Sign minus	—	Sign invalid
Unpack Unicode	Sign plus	Sign minus	—	Sign invalid
Update Tree	Compare equal at current node on path	Path complete, no nodes compared equal	—	Path not complete and compared register negative
<b>Decimal Instructions</b>				
Add Decimal	Zero	< Zero	> Zero	Overflow
Compare Decimal	Equal	First op low	First op high	—
Edit	Zero	< Zero	> Zero	—
Edit and Mark	Zero	< Zero	> Zero	—
Shift and Round Decimal	Zero	< Zero	> Zero	Overflow
Subtract Decimal	Zero	< Zero	> Zero	Overflow
Test Decimal	Digits and sign valid	Sign invalid	Digit invalid	Sign and digit invalid
Zero and Add	Zero	< Zero	> Zero	Overflow
<b>Floating-Point Instructions</b>				
Add	Zero	< Zero	> Zero	NaN
Add Normalized	Zero	< Zero	> Zero	—
Add Unnormalized	Zero	< Zero	> Zero	—
Compare (BFP)	Equal	First op low	First op high	Unordered
Compare (HFP)	Equal	First op low	First op high	—
Compare and Signal	Equal	First op low	First op high	Unordered
Convert BFP to HFP	Zero	< Zero	> Zero	Special case
Convert HFP to BFP	Zero	< Zero	> Zero	Special case
Convert to Fixed	Zero	< Zero	> Zero	Special case
Divide to Integer	Remainder complete, quotient normal	Remainder incomplete, quotient overflow or NaN	Remainder incomplete, quotient normal	Remainder incomplete, quotient overflow or NaN
Load and Test (BFP)	Zero	< Zero	> Zero	NaN
Load and Test (HFP)	Zero	< Zero	> Zero	—
Load Complement (BFP)	Zero	< Zero	> Zero	NaN
Load Complement (HFP)	Zero	< Zero	> Zero	—
Load Negative (BFP)	Zero	< Zero	—	NaN
Load Negative (HFP)	Zero	< Zero	—	—
Load Positive (BFP)	Zero	—	> Zero	NaN
Load Positive (HFP)	Zero	—	> Zero	—
Subtract	Zero	< Zero	> Zero	NaN
Subtract Normalized	Zero	< Zero	> Zero	—
Subtract Unnormalized	Zero	< Zero	> Zero	—
Test Data Class	Zero (no match)	One (match)	—	—
<b>Control Instructions</b>				
Compare and Swap and Purge	Equal	Not equal	—	—
Diagnose	See note	See note	See note	See note
Extract Stacked State	Branch state entry	Program call state entry	—	—
Insert Address Space Control	Primary-space mode	Secondary-space mode	Accessregister mode	Home-space mode

Condition Code →	0	1	2	3
Mask Bit Value →	8	4	2	1
Load Address Space Parameters	Parameters loaded	Primary not available	Secondary not authorized or not available	Space-switch event
Load PSW	See note	See note	See note	See note
Load PSW Extended	See note	See note	See note	See note
Load Real Address	Translation available	Segmentable entry invalid	Pagetable entry invalid	See note
Move to Primary	Length ≤ 256	—	—	Length > 256
Move to Secondary	Length ≤ 256	—	—	Length > 256
Move with Key	Length ≤ 256	—	—	Length > 256
Page In	Operation completed	ES data error	—	ES block not available
Page Out	Operation completed	ES data error	—	ES block not available
Program Return	See note	See note	See note	See note
Reset Reference Bit Extended	Ref = 0, Chg = 0	Ref = 0, Chg = 1	Ref = 1, Chg = 0	Ref = 1, Chg = 1
Resume Program	See note	See note	See note	See note
Set Clock	Set	Secure	—	Not operational
Signal Processor	Accepted	Status stored	Busy	Not operational
Store System Information	Info provided	—	—	Info not available
Test Access	ALET = 0	ALET uses DUALD	ALET uses PSALD	ALET = 1 or causes ART exception
Test Block	Usable	Unusable	—	—
Test Protection	Fetch and store allowed	Fetch allowed; no store allowed	No fetch or store allowed	Translation not available
<b>Input/Output Instructions</b>				
Cancel Subchannel	Function started	—	—	Not operational
Clear Subchannel	Function started	—	—	Not operational
Halt Subchannel	Function started	Nonintermediate status pending	Busy	Not operational
Modify Subchannel	Function executed	Status pending	Busy	Not operational
Reset Channel Path	Function started	—	Busy	Not operational
Resume Subchannel	Function started	Status pending	Not applicable	Not operational
Start Subchannel	Function started	Status pending	Busy	Not operational
Store Channel Report Word	CRW stored	Zeros stored	—	—
Store Subchannel	SCHIB stored	—	—	Not operational
Test Pending Interruption	Interruption not pending	Interruption code stored	—	—

<b>Condition Code →</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>Mask Bit Value →</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>

**Notes:**

For Diagnose, the resulting condition code is model-dependent.

For Load Real Address, condition code 3 is set if address-space-control element not available, region-table entry outside table or invalid, segment-table entry outside table, or, for LRA in 24- or 31-bit mode when bits 0-32 of entry address not all zeros, segment- or page-table entry invalid.

For Load PSW, Load PSW Extended, and Resume Program, the condition code is loaded from the condition-code field of the second operand.

For Set Program Mask, the condition code is loaded from bit positions 2 and 3 of the first operand.

## Assembler Instructions

Function	Mnemonic	Meaning
Option control	*PROCESS ACONTROL	Specify assembler options Dynamically modify options
Data definition	CCW CCW0 CCW1 DC DS	Define channel command word Define format-0 channel command word Define format-1 channel command word Define constant Define storage
Program sectioning and linking	ALIAS AMODE CATTR COM CSECT CXD DSECT DXD ENTRY EXTRN LOCTR RMODE RSECT START WXTRN XATTR	Rename external symbol Specify addressing mode Define class/part name and attributes Identify common control section Identify control section Cumulative length of external dummy section Identify dummy section Define external dummy section Identify entry-point symbol Identify external symbol Specify multiple location counters Specify residence mode Identify read-only control section Start assembly Identify weak external symbol Declare external symbol attributes
Base register assignment	DROP USING	Drop base address register Use base address and register
Control of listing	AEJECT ASPACE CEJECT EJECT PRINT SPACE TITLE	Start new page in macro definition Space lines in macro definition Conditional start new page Start new page Print optional data Space listing Identify assembly output
Program control	ADATA CNOP COPY END EQU EXITCTL ICTL ISEQ LTORG OPSYN ORG POP PUNCH PUSH REPRO	Provide data for SYSADATA file Conditional no operation Copy predefined source coding End assembly Equate symbol Program control data for I/O exits Input format control Input sequence checking Begin literal pool Equate operation code Set location counter Restore ACONTROL, PRINT, or USING status Punch a record Save current ACONTROL, PRINT, or USING status Reproduce following record
Conditional assembly	ACTR AGO AIF AINERT ANOP	Conditional assembly branch counter Unconditional branch Conditional branch Create input record Assembly no operation

Function	Mnemonic	Meaning
	AREAD	Assign input record to SETC symbol
	GBLA	Define global SETA symbol
	GBLB	Define global SETB symbol
	GBLC	Define global SETC symbol
	LCLA	Define local SETA symbol
	LCLB	Define local SETB symbol
	LCLC	Define local SETC symbol
	MHELP	Trace macro flow
	MNOTE	Generate error message
	SETA	Set arithmetic variable symbol
	SETAF	Set arithmetic variable symbol from external function
	SETB	Set binary variable symbol
	SETC	Set character variable symbol
	SETCF	Set character variable symbol from
Macro definition	MACRO	Macro definition header
	MEND	Macro definition trailer
	MEXIT	Macro definition exit

Source: SC26-4940.

## Extended-Mnemonic Instructions for Branch on Condition

Use	Extended Mnemonic*	Meaning	Machine Instr.* (RX or RR)
Control	B or BR NOP or NOPR	Unconditional branch No operation	BC or BCR 15, BC or BCR 0,
After Compare Instructions (A:B)	BH or BHR BL or BLR BE or BER BNH or BNHR BNL or BNLR BNE or BNER	Branch on A High Branch on A Low Branch on A Equal B Branch on A Not High Branch on A Not Low Branch on A Not Equal B	BC or BCR 2, BC or BCR 4, BC or BCR 8, BC or BCR 13, BC or BCR 11, BC or BCR 7,
After Arithmetic Instructions	BP or BPR BM or BMR BZ or BZR BO or BOR BNP or BNPR BNM or BNMR BNZ or BNZR BNO or BNOR	Branch on Plus Branch on Minus Branch on Zero Branch on Overflow Branch on Not Plus Branch on Not Minus Branch on Not Zero Branch on No Overflow	BC or BCR 2, BC or BCR 4, BC or BCR 8, BC or BCR 1, BC or BCR 13, BC or BCR 11, BC or BCR 7, BC or BCR 14,
After Test under Mask Instruction	BO or BOR BM or BMR BZ or BZR BNO or BNOR BNM or BNMR BNZ or BNZR	Branch if Ones Branch if Mixed Branch if Zeros Branch if Not Ones Branch if Not Mixed Branch if Not Zeros	BC or BCR 1, BC or BCR 4, BC or BCR 8, BC or BCR 14, BC or BCR 11, BC or BCR 7,

Source: SC26-4940.

\* Second operand, not shown, is D<sub>2</sub> (X<sub>2</sub>, B<sub>2</sub>) for RX format and R<sub>2</sub> for RR format.

## Extended-Mnemonic Instructions for Relative-Branch Instructions

Use	Extended Mnemonic	Meaning	Machine Instr.
General	BRU or J	Unconditional Branch Relative	BR 15,I <sub>2</sub>
Branch Rel. on Condition	BRUL or JLU JNOP*	Unconditional Branch Relative No Operation	BRCL 15,I <sub>2</sub> BRC 0,I <sub>2</sub>
After Compare Instructions	BRH or JH* BRL or JL* BRE or JE* BRNH or JNH* BRNL or JNL* BRNE or JNE*	Branch Relative on A High Branch Relative on A Low Branch Relative on A Equal B Branch Relative on A Not High Branch Relative on A Not Low Branch Relative on A Not Equal B	BR 2,I <sub>2</sub> BR 4,I <sub>2</sub> BR 8,I <sub>2</sub> BR 13,I <sub>2</sub> BR 11,I <sub>2</sub> BR 7,I <sub>2</sub>
After Arithmetic	BRP or JP* BRM or JM*	Branch Relative on Plus Branch Relative on Minus	BR 2,I <sub>2</sub> BR 4,I <sub>2</sub>

Use	Extended Mnemonic	Meaning	Machine Instr.
Instructions	BRZ or JZ*	Branch Relative on Zero	BRC 8,I <sub>2</sub>
	BRO or JO*	Branch Relative on Overflow	BRC 1,I <sub>2</sub>
	BRNP or JNP*	Branch Relative on Not Plus	BRC 13,I <sub>2</sub>
	BRNM or JNM*	Branch Relative on Not Minus	BRC 11,I <sub>2</sub>
	BRNZ or JNZ*	Branch Relative on Not Zero	BRC 7,I <sub>2</sub>
	BRNO or JNO*	Branch Relative on No Overflow	BRC 14,I <sub>2</sub>
After Test under Mask Instruction	BRO or JO*	Branch Relative if Ones	BRC 1,I <sub>2</sub>
	BRM or JM*	Branch Relative if Mixed	BRC 4,I <sub>2</sub>
	BRZ or JZ*	Branch Relative if Zeros	BRC 8,I <sub>2</sub>
	BRNO or JNO*	Branch Relative if Not Ones	BRC 14,I <sub>2</sub>
	BRNM or JNM*	Branch Relative if Not Mixed	BRC 11,I <sub>2</sub>
	BRNZ or JNZ*	Branch Relative if Not Zeros	BRC 7,I <sub>2</sub>
Non-Branch Relative on Condition	JAS	Branch Relative and Save	BRAS R <sub>1</sub> ,I <sub>2</sub>
	JASL	Branch Relative and Save Long	BRASL R <sub>1</sub> ,I <sub>2</sub>
	JCT	Branch Relative on Count	BRCT R <sub>1</sub> ,I <sub>2</sub>
	JCTG	Branch Relative on Count	BRCTG R <sub>1</sub> ,I <sub>2</sub>
	JXH	Branch Relative on Index High	BRXH R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>
	JXHG	Branch Relative on Index High	BRXHG R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>
JXLE	JXLE	Br. Rel. on Index Low or Equal	BRXLE R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>
	JXLEG	Br. Rel. on Index Low or Equal	BRXLG R <sub>1</sub> ,R <sub>3</sub> ,I <sub>2</sub>

Source: SC26-4940.

\* To obtain BRCL instead of BRC, add L at the end of the B mnemonic or insert L after the J of the J mnemonic. For example, change BRNZ or JNZ to BRNZL or JLNZ.

## CNOP Alignment

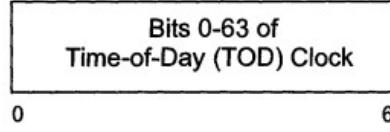
Quadword											
Doubleword						Doubleword					
Fullword		Fullword		Fullword		Fullword		Fullword		Fullword	
Halfword	Halfword	Halfword	Halfword	Halfword	Halfword	Halfword	Halfword	Halfword	Halfword	Halfword	Halfword
Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte
0,4	2,4	0,4	2,4	0,4	2,4	0,4	2,4	0,4	2,4	0,4	2,4
0,8	2,8	4,8	6,8	0,8	2,8	4,8	6,8	0,8	2,8	4,8	6,8
0,16	2,16	4,16	6,16	8,16	10,16	12,16	14,16	0,16	2,16	4,16	6,16

## Summary of Constants

Type	Implied Length, Bytes	Default Alignment	Format	Truncation/ Padding
A	4	Word	Value of address	Left
AD	8	Doubleword	Value of address	Left
B	-	Byte	Binary digits	Left
C	-	Byte	Characters	Right
CU	Even	Byte	Characters, translated to Unicode	Right
D	8	Doubleword	Long hex floating point	Right
DB	8	Doubleword	Long binary floating point	Right
DH	8	Doubleword	Long hex floating point	Right
E	4	Word	Short hex floating point	Right
EB	4	Word	Short binary floating point	Right
EH	4	Word	Short hex floating point	Right
F	4	Word	Fixed-point binary	Left
FD	8	Doubleword	Fixed-point binary	Left
G	Even	Byte	Graphic (double-byte) characters	Right
H	2	Halfword	Fixed-point binary	Left
J	4	Word	Symbol naming a DXD, DSECT, or class	Left
JD	8	Doubleword	Symbol naming a DXD, DSECT, or class	Left
L	16	Doubleword	Extended hex floating point	Right
LB	16	Doubleword	Extended binary floating point	Right
LH	16	Doubleword	Extended hex floating point	Right
LQ	16	Quadword	Extended hex floating point	Right
P	-	Byte	Packed decimal	Left
Q	4	Word	Symbol naming a DXD or DSECT	Left
QD	8	Doubleword	Symbol naming a DXD or DSECT	Left
R	4	Word	PSECT address value	Left
RD	8	Doubleword	PSECT address value	Left
S	2	Halfword	Address in base-displacement form	-
V	4	Word	Externally defined address value	-
VD	8	Doubleword	Externally defined address value	-
X	-	Byte	Hexadecimal digits	Left
Y	2	Halfword	Value of address	Left
Z	-	Byte	Zoned decimal	Left

Source: SC26-4940.

## Operand of Store Clock



Note: Bit 51 of the TOD clock corresponds to one microsecond.

## Operand of Store Clock Extended

Zeros	Time-of-Day (TOD) Clock	Programmable Field
0	8	112 127

Note: Bit 51 of the TOD clock (bit 59 of the operand) corresponds to one microsecond.

## Fixed Storage Locations

Area (Dec)	Addr Type	Hex Addr	Function
128-131	R	80	External-interruption parameter
132-133	R	84	CPU address associated with external interruption, or zeros
134-135	R	86	External-interruption code (see table on page 30)
136-139	R	88	SVC-interruption identification: 0-12 zeros, 13-14 ILC, 15 zero, 16-31 code
140-143	R	8C	Program-interruption identification: 0-12 zeros, 13-14 ILC, 15 zero, 16-31 code (see table on page 30)
144-147	R	90	Data-exception code: 0-23 zeros, 24-31 code (see table on page 32)
148-149	R	94	Monitor-class number: 0-7 zeros, 8-15 number
150-151	R	96	PER code: 0 successful branching, 1 instruction fetching, 2 storage alteration, 4 STURA (with 2), 3 and 5-6 zeros, 7 instruction-fetching nullification (with 1), 8-13 ATMID, 14-15 AI
152-159	R	98	PER address
160	R	A0	Exception access identification: 0-3 zeros, 4-7 access-register number
161	R	A1	PER access identification: 0-3 zeros, 4-7 access-register number
162	R	A2	Operand access identification (if page-translation exception recognized by MOVE PAGE): 0-3 R <sub>1</sub> , 4-7 R <sub>2</sub>
163	A/R	A3	Store-status/machine-check architectural-mode identification: 0-6 zeros, 7 one
168-175	R	A8	Translation-exception identification (see table on page 31)
176-183	R	B0	Monitor code
184-187	R	B8	Subsystem-identification word: 0-14 zeros, 15 one, 16-31 subchannel number
188-191	R	BC	I/O-interruption parameter
192-195	R	C0	I/O-interruption-identification word: 0-1 zeros, 2-4 I/O-interruption subclass, 5-31 zeros
200-203	R	C8	STFL facility list (see "Facility Indications" on page 32 for the first 32 facility bits)
232-239	R	E8	Machine-check-interruption code (see diagram on page 52)
244-247	R	F4	External-damage code (see diagram on page 52)
248-255	R	F8	Failing-storage address
272-279	R	110	Breaking-event address
288-303	R	120	Restart old PSW
304-319	R	130	External old PSW
320-335	R	140	Supervisor-call old PSW
336-351	R	150	Program old PSW
352-367	R	160	Machine-check old PSW
368-383	R	170	Input/output old PSW
416-431	R	1A0	Restart new PSW
432-447	R	1B0	External new PSW
448-463	R	1C0	Supervisor-call new PSW
464-479	R	1D0	Program new PSW
480-495	R	1E0	Machine-check new PSW
496-511	R	1F0	Input/output new PSW
4544-4607	R	11C0	Available for programming
4608-4735	A/R	1200	Store-status/machine-check floating-point-register save area
4736-4863	A/R	1280	Store-status/machine-check general-register save area
4864-4879	A/R	1300	Store-status PSW save area or machine-check fixed-logout area*
4888-4891	A	1318	Store-status prefix save area
4892-4895	A/R	131C	Store-status/machine-check floating-point-control-register save area

Area (Dec)	Addr Type	Hex Addr	Function
4900-4903	A/R	1324	Store-status/machine-check TOD-programmable-register save area
4904-4911	A/R	1328	Store-status/machine-check CPU-timer save area
4913-4919	A/R	1331	Store-status/machine-check clock-comparator bits 0-55 save area (zeros at 4912)
4928-4991	A/R	1340	Store-status/machine-check access-register save area
4992-5119	A/R	1380	Store-status/machine-check control-register save area

A Absolute address.

R Real address.

A/R A if store status; R if machine check.

\* Contents may vary among models; see System Library manuals.

## External-Interruption Codes

At real-storage locations 134-135 (86-87 hex)

Code (Hex)	Condition
0040	Interrupt key
1004	Clock comparator
1005	CPU timer
1200	Malfunction alert
1201	Emergency signal
1202	External call
1406	ETR
2401	Service signal

## Program-Interruption Codes

At real-storage locations 142-143 (8E-8F hex)

Code (Hex)	Condition
0001	Operation exception
0002	Privileged-operation exception
0003	Execute exception
0004	Protection exception
0005	Addressing exception
0006	Specification exception
0007	Data exception
0008	Fixed-point-overflow exception
0009	Fixed-point-divide exception
000A	Decimal-overflow exception
000B	Decimal-divide exception
000C	HFP-exponent-overflow exception
000D	HFP-exponent-underflow exception
000E	HFP-significance exception
000F	HFP-floating-point-divide exception
0010	Segment-translation exception
0011	Page-translation exception
0012	Translation-specification exception
0013	Special-operation exception
0015	Operand exception
0016	Trace-table exception
001C	Space-switch event
001D	HFP-square-root exception
001F	PC-translation-specification exception
0020	AFX-translation exception
0021	ASX-translation exception
0022	LX-translation exception
0023	EX-translation exception
0024	Primary-authority exception
0025	Secondary-authority exception

<b>Code (Hex)</b>	<b>Condition</b>
0026	LFX-translation exception
0027	LSX-translation exception
0028	ALET-specification exception
0029	ALEN-translation exception
002A	ALE-sequence exception
002B	ASTE-validity exception
002C	ASTE-sequence exception
002D	Extended-authority exception
002E	LSTE sequence
002F	ASTE instance
0030	Stack-full exception
0031	Stack-empty exception
0032	Stack-specification exception
0033	Stack-type exception
0034	Stack-operation exception
0038	ASCE-type exception
0039	Region-first-translation exception
003A	Region-second-translation exception
003B	Region-third-translation exception
0040	Monitor event
0080	PER event (code may be combined with another code)
0119	Crypto-operation exception

## Translation-Exception Identification

At real-storage locations 168-175 (A8-AF hex)

<b>Interruption Code (Hex)</b>	<b>Exception or Event</b>	<b>Format of Information Stored*</b>
0004	Protection	If 61 zero: rest unpredictable If 61 one: suppression, 0-51 address; if DAT was on, 60 one if access-list-controlled protection, 62-63 ASCE identification, rest unpredictable, location 160 valid; if DAT was off, rest unpredictable
0010, 0038, 0039, 003A,	Segment translation; ASCE type; Region-first translation; Region-second translation;	0-51 address, 52-61 unpredictable, 62-63 ASCE identification
003B	Region-third translation	
0011	Page translation	0-51 address, 52-60 unpredictable, if 61 zero, not MOVE PAGE; if 61 one, MOVE PAGE (see location 162); 62-63 ASCE identification
001C	Space switch	From primary-space mode: 32 old primary-space-switch-event control, 33-47 zeros, 48-63 old PASN From home-space mode: 32 home-space-switch-event control, 33-63 zeros
0020	AFX translation	32-47 zeros, 48-63 address-space number
0021	ASX translation	32-47 zeros, 48-63 address-space number
0022	LX translation	32-43 zeros, 44-63 program-call number
0023	EX translation	32-43 zeros, 44-63 program-call number
0024	Primary authority	32-47 zeros, 48-63 address-space number
0025	Secondary authority	32-47 zeros, 48-63 address-space number
0026 0027	LFX translation; LSX translation	When bit 44 is 0: 32-43 zeros, 44-63 program-call number. When bit 44 is 1, 32-63 program-call number

\* Bits 0-31 (bytes 168-171) unchanged if not described.

## Data-Exception Code (DXC)

At real-storage location 147 (93 hex) and in byte 2 of floating-point-control register

Code (Hex)	Data Exception
00	Decimal operand
01	AFP register
02	BFP instruction
08	IEEE inexact and truncated
0C	IEEE inexact and incremented
10	IEEE underflow, exact
18	IEEE underflow, inexact and truncated
1C	IEEE underflow, inexact and incremented
20	IEEE overflow, exact
28	IEEE overflow, inexact and truncated
2C	IEEE overflow, inexact and incremented
40	IEEE division by zero
80	IEEE invalid operation

## Facility Indications

Stored at real-storage locations 200-203 (C8-CB hex) by STFL; stored at second-operand location by STFLE.

Bit	Meaning when Bit is One
0	The instructions marked "N3" in the instruction-summary figures in Chapters 7 and 10 are installed.
1	The z/Architecture architectural mode is installed.
2	The z/Architecture architectural mode is active. When this bit is zero, the ESA/390 architectural mode is active.
3	The DAT-enhancement facility is installed in the z/Architecture architectural mode. The DAT-enhancement facility includes the INVALIDATE DAT TABLE ENTRY (IDTE) and COMPARE AND SWAP AND PURGE (CSPG) instructions.
4	INVALIDATE DAT TABLE ENTRY (IDTE) performs the invalidation-and-clearing operation by selectively clearing combined region-and-segment-table entries when a segment-table entry or entries are invalidated. IDTE also performs the clearing-by-ASCE operation. Unless bit 4 is one, IDTE simply purges all TLBs. Bit 3 is one if bit 4 is one.
5	INVALIDATE DAT TABLE ENTRY (IDTE) performs the invalidation-and-clearing operation by selectively clearing combined region-and-segment-table entries when a region-table entry or entries are invalidated. Bits 3 and 4 are ones if bit 5 is one.
6	The ASN-and-LX reuse facility is installed in the z/Architecture architectural mode.
7	The store-facility-list-extended facility is installed.
9	The sense-running-status facility is installed in the z/Architecture architectural mode.
16	The extended-translation facility 2 is installed.
17	The message-security assist is installed.
18	The long-displacement facility is installed in the z/Architecture architectural mode.
19	The long-displacement facility has high performance. Bit 18 is one if bit 19 is one.
20	The HFP-multiply-add/subtract facility is installed.
21	The extended-immediate facility is installed in the z/Architecture architectural mode.
22	The extended-translation facility 3 is installed in the z/Architecture architectural mode.
23	The HFP-unnormalized-extension facility is installed in the z/Architecture architectural mode.
24	The ETF2-enhancement facility is installed.
25	The store-clock-fast facility is installed in the z/Architecture architectural mode.
28	The TOD-clock-steering facility is installed in the z/Architecture architectural mode.
30	The ETF3-enhancement facility is installed in the z/Architecture architectural mode.

## Control Registers

CR	Bits	Name of Field	Associated with	Init*
0	33	SSM-suppression control	SSM instruction	0
	34	TOD-clock-sync control	TOD clock	0
	35	Low-address-protection control	Low-address protection	0
	36	Extraction-authority control	Instruction authorization	0
	37	Secondary-space control	Instruction authorization	0
	38	Fetch-protection-override control	Key-controlled protection	0
	39	Storage-protection-override control	Key-controlled protection	0
	45	AFP-register control	Floating point	0
	48	Malfunction-alert subclass mask	External interruptions	0
	49	Emergency-signal subclass mask	External interruptions	0
	50	External-call subclass mask	External interruptions	0
	52	Clock-comparator subclass mask	External interruptions	0
	53	CPU-timer subclass mask	External interruptions	0
	54	Service-signal subclass mask	External interruptions	0
	56	Unused (See note)		1
	57	Interrupt-key subclass mask	External interruptions	1
	58	Unused (See note)		1
	59	ETR subclass mask	External interruptions	0
	61	Crypto control	Cryptography	0
1	0-63	Primary address-space-control element	Dynamic address translation	0
	0-51	Primary region-table or segment-table origin or real-space token origin	Dynamic address translation	0
	54	Primary subspace-group control	Subspace groups	0
	55	Primary private-space control	Dynamic address translation	0
	56	Primary storage-alteration-event control	Program-event recording	0
	57	Primary space-switch-event control	Program interruptions	0
	58	Primary real-space control	Dynamic address translation	0
	60-61	Primary designation-type control	Dynamic address translation	0
	62-63	Primary table length	Dynamic address translation	0
2	33-57	Dispatchable-unit-control-table origin	Access-register translation	
3	32-47	PSW-key mask	Instruction authorization	
	48-63	Secondary ASN	Address spaces	
4	32-47	Authorization index	Instruction authorization	
	48-63	Primary ASN	Address spaces	
5	33-57	Primary-ASTE origin	Access-register translation	
6	32-39	I/O-interruption subclass mask	I/O interruptions	
7	0-63	Secondary address-space-control element	Dynamic address translation	0
	0-51	Secondary region-table or segment-table origin or real-space token origin	Dynamic address translation	0
	54	Secondary subspace-group control	Subspace groups	0
	55	Secondary private-space control	Dynamic address translation	0
	56	Secondary storage-alteration-event control	Program-event recording	0
	58	Secondary real-space control	Dynamic address translation	0
	60-61	Secondary designation-type control	Dynamic address translation	0

CR	Bits	Name of Field	Associated with	Init*
	62-63	Secondary table length	Dynamic address translation	0
8	32-47	Extended authorization index	Access-register translation	0
	48-63	Monitor masks	MC instruction	0
9	32	Successful-branching-event mask	Program-event recording	0
	33	Instruction-fetching-event mask	Program-event recording	0
	34	Storage-alteration-event mask	Program-event recording	0
	36	Store-using-real-address-event mask	Program-event recording	0
	39	Instruction-fetching nullification event mask	Program-event recording	0
	40	Branch-address control	Program-event recording	0
	42	Storage-alteration-space control	Program-event recording	0
	10	PER starting address	Program-event recording	0
11	0-63	PER ending address	Program-event recording	0
12	0	Branch-trace control	Tracing	0
	1	Mode-trace control	Tracing	0
	2-61	Trace-entry address	Tracing	0
	62	ASN-trace control	Tracing	0
	63	Explicit-trace control	Tracing	0
13	0-63	Home address-space-control element	Dynamic address translation	0
	0-51	Home region-table or segment-table origin or real-space token origin	Dynamic address translation	0
	54	Home subspace-group control	Subspace groups	0
	55	Home private-space control	Dynamic address translation	0
	56	Home storage-alteration-event control	Program-event recording	0
	57	Home space-switch-event control	Program interruptions	0
	58	Home real-space control	Dynamic address translation	0
	60-61	Home designation-type control	Dynamic address translation	0
	62-63	Home table length	Dynamic address translation	0
14	32	Unused (See note)		1
	33	Unused (See note)		1
	35	Channel-report-pending subclass mask	I/O machine-check handling	0
	36	Recovery subclass mask	Machine-check handling	0
	37	Degradation subclass mask	Machine-check handling	0
	38	External-damage subclass mask	Machine-check handling	1
	39	Warning subclass mask	Machine-check handling	0
	42	TOD-clock-control-override control	TOD clock	0
	44	ASN-translation control	Instruction authorization	0
15	45-63	ASN-first-table origin	ASN translation	0
	0-60	Linkage-stack-entry address	Linkage-stack operations	0

\* Value after initial CPU reset.

Note: This bit is not used but is initialized to one for consistency with the System/370 definition.

## Floating-Point-Control (FPC) Register

← Masks →   ← Flags →																					
i	z	o	u	x	0	0	0	i	z	o	u	x	0	0	DXC	0	0	0	0	0	RM
0					8									16			24				31

<u>Bit</u>	<u>Meaning</u>
0	(IMi) IEEE-invalid-operation mask
1	(IMz) IEEE-division-by-zero mask
2	(IMo) IEEE-overflow mask
3	(IMu) IEEE-underflow mask
4	(IMx) IEEE-inexact mask
8	(SFi) IEEE-invalid-operation flag
9	(SFz) IEEE-division-by-zero flag
10	(SFo) IEEE-overflow flag
11	(SFu) IEEE-underflow flag
12	(SFx) IEEE-inexact flag
16-23	(DXC) Data-exception code (see table on page 32)
30-31	(RM) Rounding mode 00 Round to nearest 01 Round toward 0 10 Round toward $+\infty$ 11 Round toward $-\infty$

## Program-Status Word (PSW)

### z/Architecture PSW

0	R	0	0	0	T	I	E	PSW Key	0	M	W	P	AS	CC	Pro-gram Mask	0	0	0	0	0	E A	
0					5				12				16	18	20		24					31

B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32	64	95
Bits 0-31 of Instruction Address		
Bits 32-63 of Instruction Address		
96		127

<u>Bit</u>	<u>Meaning</u>
1	(R) Program-event-recording mask
5	(T = 1) DAT mode
6	(I) Input/output mask
7	(E) External mask
12	Zero indicates z/Architecture
13	(M) Machine-check mask
14	(W = 1) Wait state
15	(P = 1) Problem state
16-17	xx Real mode (T = 0) 00 Primary-space mode (T = 1) 01 Access-register mode (T = 1) 10 Secondary-space mode (T = 1) 11 Home-space mode (T = 1)
18-19	(CC) Condition code
20	Fixed-point-overflow mask
21	Decimal-overflow mask
22	HFP-exponent-underflow mask
23	HFP-significance mask
31/32	Extended/basic addressing mode 00 24-bit mode 01 31-bit mode 10 Invalid 11 64-bit mode

## ESA/390 PSW

0	R	0	0	0	T	I	E	PSW Key	1	M	W	P	AS	CC	Pro- gram Mask	0	0	0	0	0	0	0	E	A	
0		5		8		12		16		18		20		24		31									

A	Instruction Address	32	63
---	---------------------	----	----

<u>Bit</u>	<u>Meaning</u>
12	One indicates ESA/390
32	(A = 1) 31-bit addressing mode

## Dynamic Address Translation

### Virtual-Address Format

← 11 →   ← 11 →   ← 11 →   ← 11 →   ← 8 →   ← 12 →
RFX      RSX      RTX      SX      PX      BX
0            11        22        33        44        52        63

RX ←————→ RX

<u>Field</u>	<u>Meaning</u>
RX	Region index (region = 2G bytes)
RFX	Region first index
RSX	Region second index
RTX	Region third index
SX	Segment index (segment = 1M bytes)
PX	Page index (page = 4K bytes)
BX	Byte index

### Address-Space-Control Element (ASCE)

#### Region-Table or Segment-Table Designation (RTD or STD)

Region-Table or Segment-Table Origin	G	P	S	X	R	DT	DL
0	52	54	58	60	63		

<u>Bit</u>	<u>Meaning</u>
54	(G) Subspace-group control
55	(P) Private-space control
56	(S) Storage-alteration-event control
57	(X) Space-switch-event control
58	(R) Real-space control (R = 0)
60-61	(DT) Designation-type control
11	Region-first-table
10	Region-second-table
01	Region-third-table
00	Segment-table
62-63	(TL) Table length ( $\times$ 4K bytes)

#### Real-Space Designation (RSD)

Real-Space Token Origin	G	P	S	X	R		
0	52	54	58	60	63		

<u>Bit</u>	<u>Meaning</u>
58	(R) Real-space control (R = 1)

Note: Other bits are as in RTD or STD.

## Table Values

Table	Incre- ment	Incr. Size	Incr. Ent- ries	Max. Size	Max. Ent- ries	Max Table Maps	
						Regions	Bytes
Region First	1-4	4KB	512	16KB	2K	8G	$16E = 16 \times 2^{60}$
Region Second	1-4	4KB	512	16KB	2K	4M	$8P = 8 \times 2^{50}$
Region Third	1-4	4KB	512	16KB	2K	2K	$4T = 4 \times 2^{40}$
Segment	1-4	4KB	512	16KB	2K	1	$2G = 2 \times 2^{30}$
Page	1	2KB	256	2KB	256	—	$1M = 2^{20}$

## Region-Table Entry (RTE)

### Region-First-Table Entry (RFTE)

Region-Second-Table Origin						TF	I	TT	TL	
0						52	56	58	60	63

### Region-Second-Table Entry (RSTE)

Region-Third-Table Origin						TF	I	TT	TL	
0						52	56	58	60	63

### Region-Third-Table Entry (RTTE)

Segment-Table Origin						TF	I	TT	TL	
0						52	56	58	60	63

#### Bit      Meaning

- 56-57 (TF) Table offset (for next-lower-level table)
- 58 (I) Invalid bit (for set of regions in RFTE or RSTE, or for region in RTTE)
- 60-61 (TT) Table-type bits (for this table)
  - 11 Region first table
  - 10 Region second table
  - 01 Region third table
- 62-63 (TL) Table length (for next-lower-level table) ( $\times 4K$  bytes)

## Segment-Table Entry (STE)

Page-Table Origin						P	I	C	TT	
0						54	58	60	63	

#### Bit      Meaning

- 54 (P) Page-protection bit
- 58 (I) Segment-invalid bit
- 59 (C) Common-segment bit
- 60-61 (TT) Table-type bits (for this table)
  - 00 Segment table

## Page-Table Entry (PTE)

Page-Frame Real Address						0	I	P	0	/	/	/	/	/	/
0						52	56		63						

#### Bit      Meaning

- 53 (I) Page-invalid bit
- 54 (P) Page-protection bit

## ASN Translation

### Address-Space Number (ASN)

ASN-First-Table Index	ASN-Second-Table Index
0	10 15

### ASN-First-Table Entry

I	ASN-Second-Table Origin	0 0 0 0 0 0
0 1		26 31

#### Bit Meaning

0 (I) AFX-invalid bit

### ASN-Second-Table Entry (ASTE)

Byte (Hex)

0	I	Authority-Table Origin	B
0 1			31

4	Authorization Index	Authority-Table Length	C R A A
0	16	28 31	

—— Address-Space-Control Element (ASCE=RTD/STD/RSD) Part 1 ——

8	Region or Segment-Table Origin or Real-Space Token Origin	0	31
---	---	---	----

—— RTD or STD Part 2 (R=0) ——

C	Region or Seg.-Table Origin (Continued)	G P S X R	D T	D L	0	20 22 26 28 31
---	---	-----------	-----	-----	---	----------------

—— RSD Part 2 (R=1) ——

C	Real-Space Token Origin (Continued)	G P S X R	0	20 22 26 31
---	-------------------------------------	-----------	---	-------------

10	Primary-Space Access-List Origin	ALL	0 1	25 31
----	----------------------------------	-----	-----	-------

14	ASN-Second-Table-Entry Sequence Number	0	31
----	--	---	----

—— Linkage-Table Designation (LTD) ——

18	V	Linkage-Table Origin	LTL	0 1	25 31
----	---	----------------------	-----	-----	-------

—— Linkage-First-Table Designation (LFTD) ——

18	V	Linkage-First-Table Origin	LFTL	0 1	24 31
----	---	----------------------------	------	-----	-------

1C	Available for Programming	0	31
----	---------------------------	---	----

#### **Byte.Bit Meaning**

- 0.0 (I) ASX-invalid bit
  - 0.31 (B) Base-space bit
  - 4.30 (CA) Controlled-ASN bit
  - 4.31 (RA) Reusable-ASN bit
  - 10.25-31 (ALL) Access-list length ( $\times$  128 bytes)
  - 18.0 (V) Subsystem-linkage control
  - 18.25-31 (LTL) Linkage-table length ( $\times$  128 bytes)
  - 18.24-31 (LFTL) Linkage-first-table length ( $\times$  256 bytes)

## **PC-Number Translation**

**Program-Call Number (20-Bit)**

	Linkage Index	Entry Index
32	44	56

## **Program-Call Number (32-Bit, Bit 44=0)**

	0	LFX	LSX	Entry Index
32	44	51	56	63

### **Program-Call Number (32-Bit, Bit 44=1)**

LFX1	1	LFX2	LSX	Entry Index
32	44	51	56	63

## **Linkage-Table Entry (LTE)**

---

I	Entry-Table Origin	ETL
0 1		26 31
Bil	Measuring	

- | <u>Bit</u> | <u>Meaning</u>                                 |
|------------|--|
| 0          | (I) LX-Invalid bit                             |
| 26-31      | (ETL) Entry-table length ( $\times$ 128 bytes) |

## **Linkage-First-Table Entry (LFTE)**

## I Linkage-Second-Table Origin

- | <u>Bit</u> | <u>Meaning</u>      |
|------------|---------------------|
| 0          | (I) LFX-invalid bit |

## Linkage-Second-Table Entry (LSTE)

I	Entry-Table Origin	ETL
0 1		26 31
LSTESN		63
32		
<b>Bit</b>	<b>Meaning</b>	
0	(I) LSX-invalid bit	
26-31	(ETL) Entry-table length ( $\times$ 128 bytes)	

## Entry-Table Entry (ETE)

Byte  
(Hex)

If Bit 10.1 (G) Is Zero

0		31
0		
4	Bits 33-62 of Entry Instruction Address	
0 1		31

If Bit 10.1 (G) Is One

0	Bits 0-31 of Entry Instruction Address	
0		31
4	Bits 32-62 of Entry Instruction Address	
0		31
8	Authorization Key Mask	
0	16	31
C	Entry Key Mask	
0	16	31
10	T G K M E C S E K	Entry Ext. Auth. Index
0 3 8 12 16		31
14	ASN-Second-Table-Entry Address	
0 1	26	31
18	Bits 0-31 of Entry Parameter	
0		31
1C	Bits 32-63 of Entry Parameter	
0		31

### Byte.Bit Meaning

- 4.0 (A) Entry addressing mode
- 4.31 (P) Entry problem state
- 10.0 (T) PC-type bit (zero: basic; one: stacking)
- 10.1 (G) Entry extended addressing mode
- 10.3 (K) PSW-key control (zero: unchanged; one: replace if stacking)
- 10.4 (M) PSW-key-mask control (zero: Or; one: replace if stacking)
- 10.5 (E) EAX control (zero: unchanged; one: replace if stacking)
- 10.6 (C) Address-space-control control
- 10.7 (S) Secondary-ASN control
- 10.8-11 (EK) Entry key

## **Access-Register Translation**

## **Access-List-Entry Token (ALET)**

0 0 0 0 0 0 0	P	ALESN	Access-List-Entry Number
0	7 8	16	31

<u>Bit</u>	<u>Meaning</u>
7	(P) Primary-list bit (zero: use DUCT; one: use primary ASTE)
8-15	(LESN) Access-list entry sequence number

## Dispatchable-Unit-Control Table (DUCT)

Byte  
(Hex)

#### In 24-Bit or 31-Bit Addressing Mode

The diagram illustrates a memory layout with two horizontal lines representing memory boundaries. The top line is labeled '20' at its left end and '31' at its right end. The bottom line is labeled '24' at its left end and '31' at its right end. Between these lines, there is a vertical white space representing memory addresses. On the far left of this vertical space, there is a small white box containing the binary digits '0 1'. To the right of this box, the text 'Bits 33-63 of Return Address' is centered.

### In 64-Bit Addressing Mode

20	Bits 0-31 of Return Address
0	31
24	Bits 32-63 of Return Address
0	31
28	
0	31

2C	Trap-Control-Block Address				I
0				29	31
30					
3C	0			31	

#### Byte.Bit Meaning

- 4.0 (SA) Subspace-active bit  
 10.25-31 (ALL) Access-list length ( $\times$  128 bytes)  
 14.28 (RA) Reduced-authority bit  
 14.31 (P) Problem-state bit  
 2C.31 (E) TRAP-enabled bit  
 /// Available for programming

### Access-List Entry (ALE)

I		F	P	ALESN	Access-List-Entry Authorization Index
0	1	6	8	16	31
32					63
ASN-Second-Table Origin					
64				90	95
96					127

Bit	Meaning
0	(I) ALEN-invalid bit
6	(FO) Fetch-only bit
7	(P) Private bit
8-15	(ALESN) Access-list-entry sequence number

### Linkage-Stack Entries

#### Entry Descriptor

U	Entry Type	Section ID	Remaining Free Space
0	1	8	16
			31
32	Next-Entry Size		
		48	63

Bit	Meaning
0	(U) Unstack-suppression bit
1-7	Entry type: Header entry = 0001001 binary Trailer entry = 0001010 binary Branch state entry = 0001100 binary Program-call state entry = 0001101 binary Available for program use = 1xxxxxx binary

### Header Entry (Entry Type 0001001)

Bits 0-31 of Backward Stack-Entry Address		
0		31
32	48	61 63
Entry Descriptor (First Half)		
64		95
Entry Descriptor (Second Half)		
96		127

Bit      Meaning

63      (B) Backward stack-entry validity bit

### Trailer Entry (Entry Type 0001010)

Bits 0-31 of Forward-Section-Header Address		
0		31
32	48	61 63
Entry Descriptor (First Half)		
64		95
Entry Descriptor (Second Half)		
96		127

Bit      Meaning

63      (F) Forward-section validity bit

### Branch State Entry (Entry Type 0001100) and Program-Call State Entry (Entry Type 0001101)

Byte  
(Hex)

0	Contents of General Registers 0-15		
78			
0			63
PSW-Key Mask Secondary ASN Ext Auth Index Primary ASN			
80	0	16	32
88	48	63	
Bits 0-63 of Program-Status Word			
0			63

In a Branch State Entry Made in 24-Bit or 31-Bit Mode

90	A	Bits 33-63 of Branch Address	
0		32	63

In a Branch State Entry Made in 64-Bit Mode

90	Bits 0-62 of Branch Address		
0			63

In a Program-Call State Entry Made on a Call to 24-Bit or 31-Bit Mode

90	Called-Space ID	0	Numeric part of PC Number
0	32	44	63

In a Program-Call State Entry Made on a Call to 64-Bit Mode

90	Called-Space ID	1	Numeric part of PC Number
0	32	44	63

98	<b>Modifiable Area</b>	63
0		63

A0	All Zeros
0	63

A8 Bits 64-127 of Program-Status Word

B0	
D8	
0	63

E0  
⋮  
118      Contents of Access Registers 0-15

120	<b>Entry Descriptor</b>	62
-----	-------------------------	----

#### **Byte.Bit Meaning**

90.32 (A) Addressing mode (in branch state entry)

### Trapping

## Trap Control Block

Byte  
(Hex)

0		P R	
0	13		31

4	
8	

0 31 C Trap-Save-Area Address

0 1 29 31  
10

14 Trap-Program Address

A horizontal number line with tick marks at integer intervals from 0 to 31. The labels are 0, 3C, ..., 20, and 31. The tick mark for 3C is positioned between 0 and 1.

## Byte.Bit Meaning

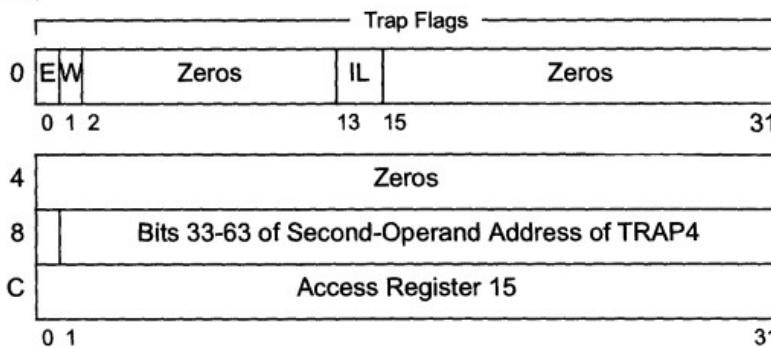
- 0.13 (P) PSW control (zero: PSW.31 must be zero, ESA/390 PSW stored; one: z/Architecture PSW stored)

0.14 (R) General-register control (zero: bits 32-63 stored; one: bits 0-63 stored)

/// Available for programming

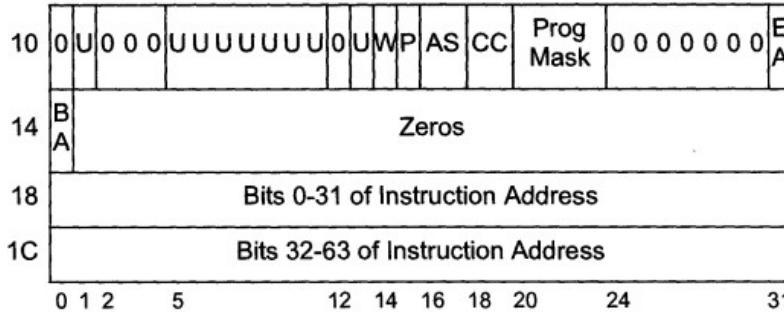
### **Trap Save Area**

Byte  
(Hex)

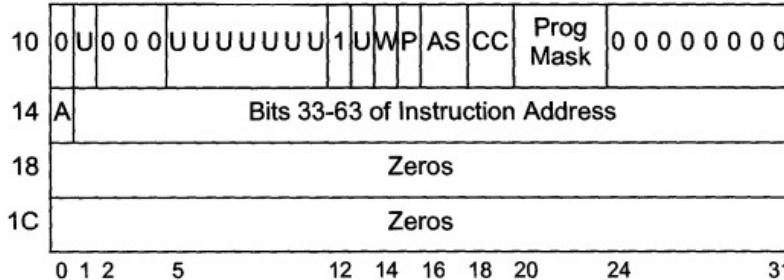


### PSW Values

### If z/Architecture PSW Stored



If ESA/390 PSW Stored



General Registers -15

**Byte.Bit Meaning**

0.0	(E) TRAP was target of EXECUTE
0.1	(W) TRAP is TRAP4 (not TRAP2)
0.13-14	(IL) Instruction-length code
10-1F	PSW values (see PSW on page 35)
U	Unpredictable
///	Available for programming

**Trace-Entry Formats****Identification of Trace Entries**

Trace-Entry Bits			Trace Entry	
0-7	8-11	12-15	Type	Format
00000000			Branch	1
00010000		000N	Set Secondary ASN	1
00100001			Program Call	1 <sup>1</sup>
00100010			Program Call	2 <sup>1</sup>
00100001		0	Program Call	3 <sup>1</sup>
00100010		0	Program Call	4 <sup>1</sup>
00100010		100E	Program Call	5 <sup>1</sup>
00100010		101E	Program Call	6 <sup>1</sup>
00100011		111E	Program Call	7 <sup>1</sup>
00110001		000N	Program Transfer	1
00110001		100N	Program Transfer	2
00110010		0000	Program Return	1
00110010		0010	Program Return	2
00110010		1000	Program Return	4
00110010		1010	Program Return	5
00110010		110N	Program Transfer	3
00110011		0011	Program Return	3
00110011		1011	Program Return	6
00110011		1100	Program Return	7
00110011		1110	Program Return	8
00110100		1111	Program Return	9
01000001			Branch in Subspace Group	1
01000010			Branch in Subspace Group	2
01010001	0010		Mode Switch	2
01010001	0011		Mode Switch	1
01010001	1010		Mode-Switching Branch	1
01010001	1011		Mode-Switching Branch	2
01010010	0110		Mode Switch	3
01010010	1100		Branch	3
01010010	1111		Mode-Switching Branch	3
0111	0		Trace	1
0111	1		Trace	2
1			Branch	

<sup>1</sup> Format-1 and -2 entries are made when the ASN-and-LX-reuse facility (ALRF) is not enabled. Entries of formats 3-7 are made when the facility is enabled.

E Indicates, when one, that the extended-addressing-mode bit, PSW bit 31, was set to one.

N Indicates, when one, that an entry was made because of PTI or SSAIR.

## Branch

F1 (Branch, RP, or TRAP2/4 to 24-Bit Mode)

00000000	Bits 40-63 of Branch Address
0      8	31

F2 (Branch, RP, or TRAP2/4 to 31/64-Bit Mode)

1	Bits 33-63 of Branch Address
0	31

F3 (Branch, RP, or TRAP2/4 to 64-Bit Mode)

010100101100	All Zeros	Bits 0-31 of Branch Address
0      8      12	32	63
Bits 32-63 of Branch Address		
64		95

Note: "Branch" is BAKR, BALR, BASR, BASSM, BSA, or BSG.

## Branch in Subspace Group (if ASN Tracing on)

F1 (in 24/31-Bit Mode)

01000001P	Bits 9-31 of ALET	A	Bits 33-63 of Branch Address
0      8	32		63

F2 (in 64-Bit Mode)

01000010P	Bits 9-31 of ALET	Bits 33-63 of Branch Address
0      8	32	63
Bits 32-63 of Branch Address		
64		95

## Mode Switch

F1 (BASSM, BSM, PC, PR, RP, or SAM64 from 24/31-Bit to 64-Bit Mode)

010100010011	All Zeros	A	Updated Instruction Address
0      8      12	32		63

F2 (BASSM, BSM, PC, PR, RP, SAM24/31 from 64-Bit to 24/31-Bit Mode)

010100010010	All Zeros	Bits 32-63 of Updated Inst. Address
0      8      12	32	63

F3 (BASSM, BSM, PC, PR, RP, SAM24/31 from 64-Bit to 24/31-Bit Mode)

010100100110	All Zeros	Bits 0-31 of Updated Inst. Address
0      8      12	32	
Bits 32-63 of Updated Inst. Address		
64		95

## Mode-Switching Branch

F1 (BASSM or RP from 64-Bit to 24/31-Bit Mode)

01010001	1010	All Zeros	A	Branch Address	
0	8	12	32		63

F2 (BASSM or RP from 24/31-Bit to 64-Bit Mode)

01010001	1011	All Zeros		Bits 32-63 of Branch Address	
0	8	12	32		63

F3 (BASSM or RP from 24/31-Bit to 64-Bit Mode)

01010010	1111	All Zeros		Bits 0-31 of Branch Address	
0	8	12	32		63
64		Bits 32-63 of Branch Address	95		

## Program Call

F1 (in 24/31-Bit Mode, ALRF Not Enabled)

00100001	PSW Key	All Zeros	A	Bits 33-62 of Return Address	P
0	8	12	32		63

F2 (in 64-Bit Mode, ALRF Not Enabled)

00100010	PSW Key	All Zeros		Bits 0-31 of Return Address	
0	8	12	32		63
64		Bits 32-62 of Return Address	95	P	

F3 (in 24/31-Bit Mode, ALRF Enabled, 20-Bit PC Number)

00100001	PSW Key	0	Bits 1-19 of 20-Bit PC Number	A	Bits 33-62 of Return Address	P
0	8	12	32		63	

F4 (in 64-Bit Mode, ALRF Enabled, 20-Bit PC Number)

00100010	PSW Key	0	Bits 1-19 of 20-Bit PC Number		Bits 0-31 of Return Address	
0	8	12	32		63	
64		Bits 32-62 of Return Address	95	P		

F5 (in 24/31-Bit Mode, ALRF Enabled, 32-Bit PC Number)

00100010	PSW Key	100E	All Zeros	A	Bits 33-62 of Return Address	P
0	8	12	16	32		63
64		32-Bit PC Number	95			

F6 (in 64-Bit Mode, ALRF Enabled, 32-Bit PC Number, Bits 0-31 of Return Address All Zeros)

00100010	PSW Key	101E	All Zeros	Bits 32-62 of Return Address	P
0	8	12	16	32	63
32-Bit PC Number					95
64					96

F7 (in 64-Bit Mode, ALRF Enabled, 32-Bit PC Number, Bits 0-31 of Return Address Not All Zeros)

00100011	PSW Key	111E	All Zeros	Bits 0-31 of Return Address	
0	8	12	16	32	63
Bits 32-62 of Return Address					P
64					32-Bit PC Number
					96
					127

## Program Return

F1 (in 24/31-Bit to 24/31-Bit Mode)

00110010	PSW Key	0000	New PASN	A	Bits 33-62 of Return Address	P
0	8	12	16	32		63
A Bits 33-63 of Updated Inst. Address						95
64						

F2 (in 64-Bit to 24/31-Bit Mode)

00110010	PSW Key	0010	New PASN	A	Bits 33-62 of Return Address	P
0	8	12	16	32		63
Bits 32-63 of Updated Inst. Address						95
64						

F3 (in 64-Bit to 24/31-Bit Mode)

00110011	PSW Key	0011	New PASN	A	Bits 33-62 of Return Address	P
0	8	12	16	32		63
Updated Instruction Address						127
64						

F4 (in 24/31-Bit to 64-Bit Mode)

00110010	PSW Key	1000	New PASN		Bits 32-62 of Return Address	P
0	8	12	16	32		63
A Bits 33-63 of Updated Inst. Address						95
64						

**F5 (in 64-Bit to 64-Bit Mode)**

00110010	PSW Key	1010	New PASN	Bits 32-62 of Return Address	P
0	8	12	16	32	63
Bits 32-63 of Updated Inst. Address					
64				95	

**F6 (in 64-Bit to 64-Bit Mode)**

00110011	PSW Key	1011	New PASN	Bits 32-62 of Return Address	P
0	8	12	16	32	63
Updated Instruction Address					
64				127	

**F7 (in 24/31-Bit to 64-Bit Mode)**

00110011	PSW Key	1100	New PASN	Bits 0-31 of Return Address	
0	8	12	16	32	63
Bits 32-62 of Return Address					P
64				96	127

**F8 (in 64-Bit to 64-Bit Mode)**

00110011	PSW Key	1110	New PASN	Bits 0-31 of Return Address	
0	8	12	16	32	63
Bits 32-62 of Return Address					P
64				96	127

**F9 (in 64-Bit to 64-Bit Mode)**

00110100	PSW Key	1111	New PASN	Bits 0-31 of Return Address	
0	8	12	16	32	63
Bits 32-62 of Return Address					P
64				96	127
Bits 32-63 of Updated Inst. Address					
128				159	

### Program Transfer

**F1 (in 24/31-Bit Mode)**

00110001	PSW Key	000N	New PASN	Bits 32-63 of R <sub>2</sub> Before	
0	8	12	16	32	63

**F2 (in 64-Bit Mode, Bits 0-31 of R<sub>2</sub> All Zeros)**

00110001	PSW Key	100N	New PASN	Bits 32-63 of R <sub>2</sub> Before	
0	8	12	16	32	63

| F3 (in 64-Bit Mode, Bits 0-31 of R<sub>2</sub> Not All Zeros)

00110001	PSW Key	100N	New PASN	Bits 0-31 of R <sub>2</sub> Before
0	8	12	16	32
Bits 32-63 of R <sub>2</sub> Before				63
64			95	

**Set Secondary ASN**

F1

00010000	0000000N	New SASN
0	8	16

31

**Trace**

F1 (TRACE)

0111	N	00000000	TOD-Clock Bits 16-63
0	8	12	16
TRACE Operand			(R <sub>1</sub> ) - (R <sub>3</sub> )
64		96	4N+16

F2 (TRACG)

0111	N	10000000	TOD-Clock Bits 0-47
0	8	12	16
TOD-Clock Bits 48-79			TRACE Operand
64		96	127
(R <sub>1</sub> ) - (R <sub>3</sub> )			8N+24
128			

**Bit      Meaning**

4-7    (N) One less than the number of registers in the trace entry.

## Machine-Check Interruption Code

At real-storage locations 232-239 (E8-EF hex)

S	P	S	C	E	D	G	C	S	C		B	S	S	K	D	W	M	P	I	F	E	F	G	C	S			
D	D	R	0	D	D	0	W	P	P	K	0	0	B	0	E	C	E	S	P	S	M	A	0	C	P	R	0	T
0	4	8					14	16					24	26												31		

I	A	D			P	F	A		C	C																	
E	R	A	0	0	0	0	0	0	R	C	P	0	T	C	0	0	0	0	0	0	0	0	0	0	0	0	0
32	35		40	42		46	48																			56	63

### Bit      Meaning

- 0      (SD) System damage
- 1      (PD) Instruction-processing damage
- 2      (SR) System recovery
- 4      (CD) Timing-facility damage
- 5      (ED) External damage
- 7      (DG) Degradation
- 8      (W) Warning
- 9      (CP) Channel report pending
- 10     (SP) Service-processor damage
- 11     (CK) Channel-subsystem damage
- 14     (B) Backed up
- 16     (SE) Storage error uncorrected
- 17     (SC) Storage error corrected
- 18     (KE) Storage-key error uncorrected
- 19     (DS) Storage degradation
- 20     (WP) PSW-MWP validity
- 21     (MS) PSW mask and key validity
- 22     (PM) PSW program-mask and condition-code validity
- 23     (IA) PSW-instruction-address validity
- 24     (FA) Failing-storage-address validity
- 26     (EC) External-damage-code validity
- 27     (FP) Floating-point-register validity
- 28     (GR) General-register validity
- 29     (CR) Control-register validity
- 31     (ST) Storage logical validity
- 32     (IE) Indirect storage error
- 33     (AR) Access-register validity
- 34     (DA) Delayed-access exception
- 42     (PR) TOD-programmable-register validity
- 43     (FC) Floating-point-control-register validity
- 44     (AP) Ancillary report
- 46     (CT) CPU-timer validity
- 47     (CC) Clock-comparator validity

## External-Damage Code

At real-storage address 244-247 (F4-F7 hex)

0	0	0	0	0	0	0	X	X	N	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0							8	10			16														24	31

### Bit      Meaning

- 8      (XN) Expanded storage not operational
- 9      (XF) Expanded-storage control failure

## Operation-Request Block (ORB)

Word

0	Interruption Parameter																											
1	Key	S	C	M	Y	F	P	I	A	U	O	H	T	LPM	L	D	0 0 0 0 0 X											
2	0	Channel-Program Address																										
3	CSS Priority				Reserved				CU Priority				Reserved															
4	Reserved																											
5	Reserved																											
6	Reserved																											
7	Reserved																											
	0													24														
														31														

**Word.Bit      Meaning**

1.0-3	(Key) Subchannel key
1.4	(S) Suspend control
1.5	(C) Streaming-mode control
1.6	(M) Modification control
1.7	(Y) Synchronization control
1.8	(F) CCW-format control
1.9	(P) Prefetch control
1.10	(I) Initial-status-interruption control
1.11	(A) Address-limit-checking control
1.12	(U) Suppress-suspended-interruption control
1.14	(H) Format-2-IDAW control
1.15	(T) 2K-IDAW control
1.16-23	(LPM) Logical-path mask
1.24	(L) Incorrect-length-suppression mode
1.25	(D) Modified-CCW-indirect-data-addressing control
1.31	(X) ORB-extension control
3.0-7	Channel-subsystem priority
3.16-23	Control-unit priority

## Channel-Command Word (CCW)

### Format-0 CCW

Command Code	Data Address															
0																
	Flags												Byte Count			
32																

**Bit      Meaning**

32	(CD) Causes use of data-address portion of next CCW
33	(CC) Causes use of command code and data address of next CCW
34	(SLI) Causes suppression of possible incorrect-length indication
35	(Skip) Suppresses transfer of information to main storage
36	(PCI) Causes an intermediate-interruption condition to occur
37	(IDA) Causes bits 8-31 of CCW to specify location of first IDAW
38	(Suspend) Causes suspension before execution of this CCW
39	(MIDA) Causes bits 8-31 of CCW to specify location of first MIDA

## Format-1 CCW

Command Code	Flags	Byte Count	
0	8	16	31
0	Data Address		
32			

Bit	Meaning
8	(CD) Causes use of data-address portion of next CCW
9	(CC) Causes use of command code and data address of next CCW
10	(SLI) Causes suppression of possible incorrect-length indication
11	(Skip) Suppresses transfer of information to main storage
12	(PCI) Causes an intermediate-interruption condition to occur
13	(IDAW) Causes bits 33-63 of CCW to specify location of first IDAW
14	(Suspend) Causes suspension before execution of this CCW
15	(MIDAW) Causes bits 33-63 of CCW to specify location of first MIDAW

## Indirect-Data-Address Word (IDAW)

### Format-1 IDAW

0	Data Address		
1			

### Format-2 IDAW

0	Bits 0-31 of Data Address		
1			
32	Bits 32-63 of Data Address		
33			

## Modified-CCW-Indirect-Data-Address Word (MIDAW)

0	Reserved		
32	40	48	63
33	Bits 0-31 of Data Address		
34			
35	Bits 32-63 of Data Address		
36			

Bit	Meaning
40	Last MIDAW
41	Skip
42	Data-transfer-interruption control
43-47	Reserved

## Subchannel-Information Block (SCHIB)

Word

0	
1	
2	
3	Path-Management-Control Word
4	
5	
6	
7	
8	Subchannel-Status Word (See "Subchannel-Status Word (SCSW)" on page 56.)
9	
10	Model-Dependent Area / Measurement-Block Address
11	
12	Model-Dependent Area

0

31

## Path-Management-Control Word (PMCW)

Word

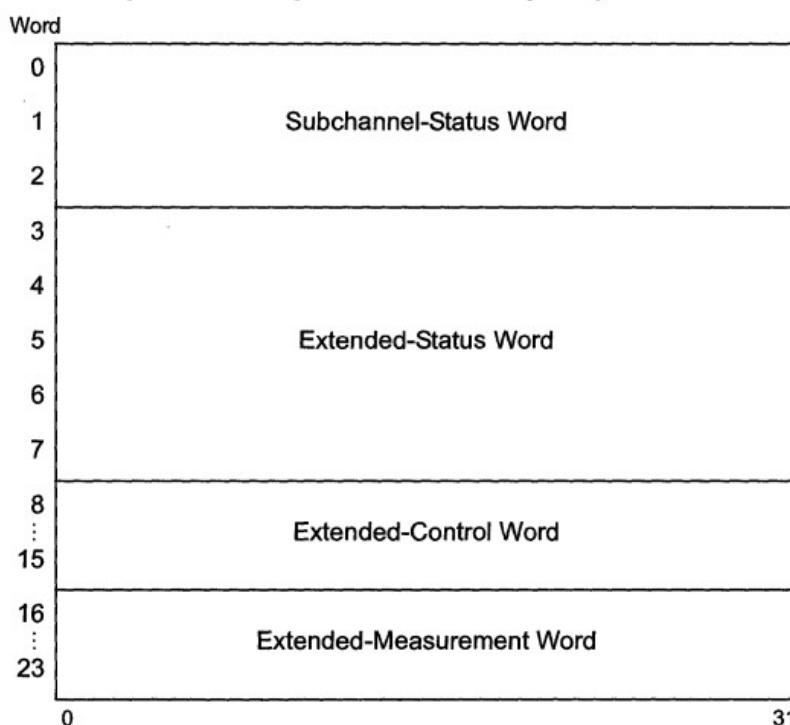
Interruption Parameter																
0	0	0	ISC	0	0	0	E	L	M	M	D	T	V			Device Number
1	0	0	LPM	PNOM		LPUM		PIM								
2	MBI		POM		POM		PAM									
3	CHPID-0		CHPID-1		CHPID-2		CHPID-3									
4	CHPID-4		CHPID-5		CHPID-6		CHPID-7									
5	0		0		0		0		F		X		S			
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	8		16		24									31	

### Word.Bit Meaning

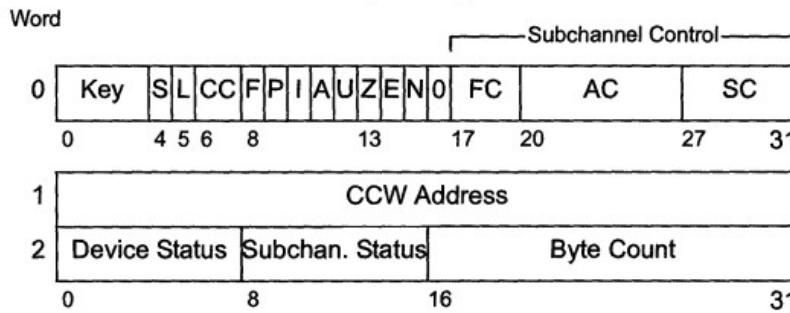
- 1.2-4 (ISC) Interruption-subclass code
- 1.8 (E) Subchannel enabled
- 1.9-10 (LM) limit mode
  - 00 No Checking
  - 01 Data address must be  $\geq$  limit
  - 10 Data address must be  $\leq$  limit
  - 11 Reserved
- 1.11-12 (MM) Measurement-mode enable
  - 00 Neither mode enabled
  - 01 Device-connect-time-measurement enabled
  - 10 Measurement-block-update enabled
  - 11 Both modes enabled

1.13	(D) Multipath mode
1.14	(T) Timing facility available
1.15	(V) Device number valid
2.0-7	(LPM) Logical-path mask
2.8-15	(PNOM) Path-not-operational mask
2.16-23	(LPUM) Last-path-used mask
2.24-31	(PIM) Path-installed mask
3.0-15	(MBI) Measurement-block index
3.16-23	(POM) Path-operational mask
3.24-31	(PAM) Path-available mask
4.0-7	(CHPID-0) Channel-path ID for logical path 0 (typical)
6.29	(F) Measurement-block-format control
6.30	(X) Extended-measurement-word-mode enable
6.31	(S) Concurrent sense

### **Interruption-Response Block (IRB)**



## **Subchannel-Status Word (SCSW)**



<b>Word.Bit</b>	<b>Meaning</b>
0.0-3	(Key) Subchannel key
0.4	(S) Suspend control
0.5	(L) Extended-status-word format (logout stored)
0.6-7	(CC) Deferred condition code <ul style="list-style-type: none"> <li>00 Normal I/O interruption</li> <li>01 Status in SCSW</li> <li>10 Reserved</li> <li>11 Path not operational</li> </ul>
0.8	(F) CCW-format control
0.9	(P) Prefetch control
0.10	(I) Initial-status-interruption control

0.11	(A) Address-limit-checking control
0.12	(U) Suppress-suspended-interruption control
0.13	(Z) Zero condition code
0.14	(E) Extended control (information stored in ECW of IRB)
0.15	(N) Path not operational (PNOM nonzero)
0.17-19	(FC) Function control 17 (40) Start, 18 (20) Halt, 19 (10) Clear
0.20-26	(AC) Activity control 20 (08) Resume pending      24 (80) Subchannel active 21 (04) Start pending      25 (40) Device active 22 (02) Halt pending      26 (20) Suspended 23 (01) Clear pending
0.27-31	(SC) Status control 27 (10) Alert      30 (02) Secondary 28 (08) Intermediate      31 (01) Status pending 29 (04) Primary
2.0-15	Device status (0-7)      Subchannel status (8-15) 0 (80) Attention      8 (80) Program-controlled int. 1 (40) Status modifier      9 (40) Incorrect length 2 (20) Control-unit end      10 (20) Program check 3 (10) Busy      11 (10) Protection check 4 (08) Channel end      12 (08) Channel-data check 5 (04) Device end      13 (04) Channel-control check 6 (02) Unit check      14 (02) Interface-control check 7 (01) Unit exception      15 (01) Chaining check

## Extended-Status Word (ESW)

See chart on page 58 to determine the appropriate ESW format.

### Format-0 ESW

Word

0	Subchannel Logout
1	Extended-Report Word
2	Failing-Storage Address
3	
4	Secondary-CCW Address

0

31

### Format-0 ESW Word 0 (Subchannel Logout)

Bit	Meaning
1-7	(ESF) Extended-status flags (1 key check, 2 measurement-block program check, 3 measurement-block data check, 4 measurement-block protection check, 5 CCW check, 6 IDAW check, 7:0)
8-15	(LPUM) Last-path-used mask
17-21	(FVF) Field-validity flags (17 LPUM, 18 TC, 19 SC, 20 device status, 21 CCW address)
22-23	(SA) Storage-access code (00 access type unknown, 01 read, 10 write, 11 read backward)
24-25	(TC) Termination code (00 halt signal issued, 01 stop, stack, or normal termination, 10 clear signal issued)
26	(D) Device status check
27	(E) Secondary error
28	(A) I/O-error alert
29-31	(SC) Sequence code

## **Format-0 ESW Word 1 (Extended-Report Word)**

<u>Bit</u>	<u>Meaning</u>
1	(L) Request logging only
2	(E) Extended-subchannel-logout pending
3	(A) Authorization check
4	(P) Path-verification-required
5	(T) Channel-path timeout
6	(F) Failing-storage-address validity
7	(S) Concurrent sense
8	(C) Secondary-CCW-address validity
9	(R) Failing-storage-address format (zero: 1-31 of word 2; one: words 2 and 3)
10-15	(SCNT) Concurrent-sense count

Format-1 ESW Word 07

**Bit**      **Meaning**  
8-15      (LPUM) Last-path-used mask

Format-2 ESW Word 01

0 0 0 0 0 0 0 0	LPUM	DCTI
0	8	16

<u>Bit</u>	<u>Meaning</u>
8-15	(LPUM) Last-path-used mask
16-31	(DCTI) Device-connect-time interval

Format-3 ESW Word 07

0 0 0 0 0 0 0 0	LPUM	Unpredictable
0	8	16

**Bit**      **Meaning**  
8-15      (LPUM) Last-path-used mask

### **Information Stored in ESW**

Subchannel Conditions under which ESW Is Stored by Test Subchannel Instruction							Extended-Status Word (ESW)						
Subchannel-Status Word			Path-Management Control Word		Device-Connect-Time Measurement-Mode Active	Device-Connect-Time Measurement-Mode Enable Bit	Timing-Facility Bit	Suspended Bit	Status-Control Field	Contents Word 0 Byte			
A	I	P	S	X						Format	0	1	2
- - - - 0		-	*	*	*	*	No / Yes	U		*	*	*	*
* * 0 0 1		1	*	*	*	*	No / Yes	0	R R R R				
* * 1 * 1		1	*	*	*	*	No / Yes	0	R R R R				
1 0 0 1 1		1	*	*	*	*	No / Yes	0	R R R R				
0 0 0 0 1		0	*	*	*	*	No / Yes	U		*	*	*	*
0 0 0 1 1		0	*	*	*	*	No / Yes	3	Z M * *				
1 0 0 * 1		0	*	*	*	*	No / Yes	3	Z M * *				
* * 1 * 1		0	*	0	*	No / Yes	1	Z M Z Z					

1 Word 1 is the same as word 1 of a format-0 ESW. Words 2, 3, and 4 are zeros.

Subchannel Conditions under which ESW Is Stored by Test Subchannel Instruction							Extended-Status Word (ESW)	
Subchannel-Status Word			Path-Management-Control Word		Device-Connect-Time Measurement-Mode Enable Bit	Device-Connect-Time Measurement-Mode Active		
Status-Control Field		Suspended Bit	Timing-Facility Bit	Device-Connect-Time Measurement-Mode Active		Format	Contents Word 0 Byte	
A	I	P	S	X	L Bit			0 1 2 3
*	*	1	*	1	0	*	1	0
*	*	1	*	1	0	*	1	1
*	*	1	*	1	0	*	1	2
0	1	0	0	1	0	*	*	Z M Z Z
0	1	0	0	1	0	*	*	Z M Z Z
0	1	0	0	1	0	*	*	Z M D D
0	1	0	0	1	0	*	*	Z M D D
0	1	0	0	1	0	*	*	Z M Z Z
0	1	0	0	1	0	*	*	Z M Z Z
0	1	0	0	1	0	*	*	Z M Z Z
0	0	0	1	1	1	1	1	Z M D D
1	1	0	0	1	1	1	2	Z M D D
*	1	0	1	1	*	These combinations do not occur.		

**Bit Meaning**

- Not meaningful.
- \*
 Bits may be zeros or ones.
 

  - A Alert status.
  - D Accumulated device-connect-time-interval (DCTI) value stored in bytes 2 and 3.
  - I Intermediate status.
  - L Extended-status-word format.
  - M Last-path-used mask (LPUM) stored in byte 1.
  - P Primary status.
  - R Subchannel-logout information stored in bytes 0-3.
  - S Secondary status.
  - U No format defined.
  - X Status pending.
  - Z Bits are stored as zeros.

### Extended-Control Word (ECW)

SCSW Bits		ERW Bit 7	ERW Bits 10-15	ECW Words 0-7
5	14			
0	0	0	Zeros	Unpredictable
0	1	1	Number of concurrent-sense bytes <sup>a</sup>	Concurrent-sense information <sup>a</sup>
1	0	0	Zeros	Unpredictable
1	1	0	Zeros	Model-dependent information
1	1	1	Number of concurrent-sense bytes	Concurrent-sense information

- a The contents of the ECW are specified by bits 5 and 14 of word 0 of the SCSW. The combination of SCSW bit 5 zero, SCSW bit 14 one, and ERW bit 7 zero does not occur.

## Extended-Measurement Word

Word

0	Device-Connect Time
1	Function-Pending Time
2	Device-Disconnect Time
3	Control-Unit-Queuing Time
4	Device-Active-Only Time
5	Device-Busy Time
6	Initial-Command-Response Time
7	Reserved

0

31

## Format 0 Measurement Block

Word

0	SSCH + RSCH Count	Sample Count
1	Device-Connect Time	
2	Function-Pending Time	
3	Device-Disconnect Time	
4	Control-Unit-Queuing Time	
5	Device-Active-Only Time	
6	Device-Busy Time	
7	Initial-Command-Response Time	

0

16

31

## Format 1 Measurement Block

Word

0	SSCH + RSCH Count
1	Sample Count
2	Device-Connect Time
3	Function-Pending Time
4	Device-Disconnect Time
5	Control-Unit-Queuing Time
6	Device-Active-Only Time
7	Device-Busy Time
8	Initial-Command-Response Time
9	Reserved
15	

0

31

## Channel-Report Word (CRW)

0	S	R	C	RSC	A	0	ERC		Reporting-Source ID	
0		4		8	10		16			31

Bit	Meaning
1	(S) Solicited CRW
2	(R) Overflow (one or more CRWs lost)
3	(C) Chaining (meaningless if bit 2 is one)
4-7	(RSC) Reporting-source code (see Reporting-Source table)
8	(A) Ancillary report
10-15	(ERC) Error-recovery code (see Error-Recovery-Code table)
16-31	Reporting-source ID (see Reporting-Source table)

## Error-Recovery Codes

ERC	Condition
0 0 0 0 0 1	Available
0 0 0 0 1 0	Initialized
0 0 0 0 1 1	Temporary error
0 0 0 1 0 0	Installed parameters initialized
0 0 0 1 0 1	Terminal
0 0 0 1 1 0	Permanent error with facility not initialized
0 0 0 1 1 1	Permanent error with facility initialized
0 0 1 0 0 0	Installed parameters modified

## Reporting Source

The reporting-source-ID format depends on the RSC field of the channel-report word, as follows:

RSC	Reporting Source	Reporting-Source ID
0 0 1 0	Monitoring facility	0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1	Subchannel	X X X X X X X X X X X X X X X X
0 1 0 0	Channel path	0 0 0 0 0 0 0 0 0 Y Y Y Y Y Y Y Y Y Y
1 0 0 1	Configuration-alert facility	0 0 0 0 0 0 0 0 0 Y Y Y Y Y Y Y Y Y Y
1 0 1 1	Channel subsystem	0 0

X = Subchannel number

Y = Channel-path ID (CHPID)

## I/O Command Codes

### Standard Command-Code Assignments (CCW Bits 0-7)

xxxx 0000	Invalid Command	mmmm 0100	Sense
mmmm mm01	Write	0000 0100	— Basic Sense
mmmm mm10	Read	1110 0100	— Sense ID
0000 0010	— Read IPL	x x x 1000	Transfer in channel (a)
mmmm mm11	Control	0000 1000	Transfer in channel (b)
0000 0011	— Control no operation	mmmm 1000	Invalid command (c)
		mmmm 1100	Read backwards
x – Bit Ignored		a	Format-0 CCW
m – Modifier bit for specific type of I/O device		b	Format-1 CCW
		c	Format-1 CCW and nonzero m bit

### Standard Meanings of Bits of First Sense Byte

Bit	Designation	Bit	Designation
0	Command reject	4	Data check
1	Intervention required	5	Overrun
2	Bus-out check	6	(Device dependent)
3	Equipment check	7	(Device dependent)

## Code Assignments

Dec	Hex	EBCDIC <sup>1</sup>	ISO-8 <sup>2</sup>
0	00	NUL	NUL
1	01	SOH	SOH
2	02	STX	STX
3	03	ETX	ETX
4	04	SEL	EOT
5	05	HT	ENQ
6	06	RNL	ACK
7	07	DEL	BEL
8	08	GE	BS
9	09	SPS	HT
10	0A	RPT	LF
11	0B	VT	VT
12	0C	FF	FF
13	0D	CR	CR
14	0E	SO	SO
15	0F	SI	SI
16	10	DLE	DLE
17	11	DC1	DC1
18	12	DC2	DC2
19	13	DC3	DC3
20	14	RES/ENP	DC4
21	15	NL	NAK
22	16	BS	SYN
23	17	POC	ETB
24	18	CAN	CAN
25	19	EM	EM
26	1A	UBS	SUB
27	1B	CU1	ESC
28	1C	IFS	IFS
29	1D	IGS	IGS
30	1E	IRS	IRS
31	1F	ITB/IUS	IUS
32	20	DS	SP
33	21	SOS	!
34	22	FS	"
35	23	WUS	#
36	24	BYP/INP	\$
37	25	LF	%
38	26	ETB	&
39	27	ESC	'
40	28	SA	(
41	29	SFE	)
42	2A	SM/SW	*
43	2B	CSP	+
44	2C	MFA	,
45	2D	ENQ	-
46	2E	ACK	.
47	2F	BEL	/
48	30		0
49	31		1
50	32	SYN	2
51	33	IR	3
52	24	PP	4
53	35	TRN	5
54	36	NBS	6
55	37	EOT	7
56	38	SBS	8
57	39	IT	9
58	3A	RFF	:
59	3B	CU3	:
60	3C	DC4	<
61	3D	NAK	=
62	3E		>
63	3F	SUB	?

Dec	Hex	EBCDIC <sup>1</sup>	ISO-8 <sup>2</sup>
64	40	SP	@
65	41	RSP	A
66	42	â	B
67	43	ä	C
68	44	à	D
69	45	á	E
70	46	ã	F
71	47	å	G
72	48	ç	H
73	49	ñ	I
74	4A	¢	J
75	4B	.	K
76	4C	<	L
77	4D	(	M
78	4E	+	N
79	4F	-	O
80	50	&	P
81	51	é	Q
82	52	ê	R
83	53	ë	S
84	54	è	T
85	55	í	U
86	56	î	V
87	57	ï	W
88	58	í	X
89	59	ß	Y
90	5A	!	Z
91	5B	\$	[
92	5C	*	\
93	5D	)	]
94	5E	;	^
95	5F	¬	-
96	60	-	-
97	61	/	a
98	62	Å	b
99	63	Ä	c
100	64	À	d
101	65	Á	e
102	66	Ã	f
103	67	Ä	g
104	68	ç	h
105	69	ñ	i
106	6A	!	j
107	6B	,	k
108	6C	%	l
109	6D	-	m
110	6E	>	n
111	6F	?	o
112	70	ø	p
113	71	É	q
114	72	É	r
115	73	É	s
116	74	È	t
117	75	Í	u
118	76	Î	v
119	77	Ï	w
120	78	-	x
121	79	,	y
122	7A	:	z
123	7B	#	{
124	7C	@	-
125	7D	'	}
126	7E	=	~
127	7F	"	-

Dec	Hex	EBCDIC <sup>1</sup>	ISO-8 <sup>2</sup>
128	80	Ø	
129	81	a	
130	82	b	BPH
131	83	c	NBH
132	84	d	IND
133	85	e	NEL
134	86	f	SSA
135	87	g	ESA
136	88	h	HTS
137	89	i	HTJ
138	8A	«	VTS
139	8B	»	PLD
140	8C	ø	PLU
141	8D	ý	RI
142	8E	þ	SS2
143	8F	±	SS3
144	90	°	DCS
145	91	j	PU1
146	92	k	PU2
147	93	l	STS
148	94	m	CCH
149	95	n	MW
150	96	o	SPA
151	97	p	EPA
152	98	q	SOS
153	99	r	
154	9A	ª	SCI
155	9B	º	CSI
156	9C	æ	ST
157	9D	·	OSC
158	9E	Æ	PM
159	9F	¤	APC
160	A0	µ	RSP
161	A1	~	i
162	A2	s	¢
163	A3	t	£
164	A4	u	¤
165	A5	v	¥
166	A6	w	-
167	A7	x	§
168	A8	y	"
169	A9	z	©
170	AA	i	a
171	AB	¿	«
172	AC	Đ	¬
173	AD	Ý	SHY
174	AE	þ	®
175	AF	®	-
176	B0	^	°
177	B1	£	±
178	B2	¥	²
179	B3	.	³
180	B4	©	'
181	B5	§	µ
182	B6	¶	¶
183	B7	¼	.
184	B8	½	,
185	B9	¾	1
186	BA	[	o
187	BB	]	»
188	BC	ä	¼
189	BD	„	½
190	BE	ſ	¾
191	BF	×	¿

Dec	Hex	EBCDIC <sup>1</sup>	ISO-8 <sup>2</sup>
192	C0	{	À
193	C1	A	Á
194	C2	B	Â
195	C3	C	Ã
196	C4	D	À
197	C5	E	Á
198	C6	F	Æ
199	C7	G	Ç
200	C8	H	É
201	C9	I	É
202	CA	SHY	Ê
203	CB	ô	Ê
204	CC	ö	Í
205	CD	ò	Í
206	CE	ó	Í
207	CF	õ	Í
208	D0	}	Ð
209	D1	J	Ñ
210	D2	K	Ó
211	D3	L	Ó
212	D4	M	Ó
213	D5	N	Õ
214	D6	O	Ó
215	D7	P	×
216	D8	Q	Ø
217	D9	R	Ú
218	DA	ı	Ú
219	DB	ú	Ú
220	DC	ü	Ü
221	DD	ù	Ý
222	DE	ú	Þ
223	DF	ÿ	Þ
224	E0	\	à
225	E1	+	á
226	E2	S	â
227	E3	T	ã
228	E4	U	ä
229	E5	V	á
230	E6	W	æ
231	E7	X	ç
232	E8	Y	é
233	E9	Z	é
234	EA	²	ê
235	EB	Ô	ë
236	EC	Ö	í
237	ED	Ó	í
238	EE	Ó	í
239	EF	Ö	í
240	F0	0	ð
241	F1	1	ñ
242	F2	2	ò
243	F3	3	ó
244	F4	4	ð
245	F5	5	ð
246	F6	6	ö
247	F7	7	+
248	F8	8	ø
249	F9	9	ù
250	FA	³	ú
251	FB	Û	ú
252	FC	Ù	ü
253	FD	Ú	ý
254	FE	Ú	þ
255	FF	EO	ý

**Notes:**

- The EBCDIC characters are based on code page 037.
- The ISO-8 controls are from ISO 6429, and the graphics are from ISO 8859-1. The ISO-8 graphics are code page 00819, named ISO/ANSI Multilingual.

**Control Character Representations**

ACK	Acknowledge	IT	Indent Tab
BEL	Bell	ITB	Intermediate Transmission Block
BS	Backspace	IUS	International Unit Separator
BYP	Bypass	LF	Line Feed
CAN	Cancel	MFA	Modify Field Attribute
CR	Carriage Return	NAK	Negative Acknowledge
CSP	Control Sequence Prefix	NBS	Numeric Backspace
CU1	Customer Use 1	NL	New Line
CU3	Customer Use 3	NUL	Null
DC1	Device Control 1	POC	Program-Operator Communication
DC2	Device Control 2	PP	Presentation Position
DC3	Device Control 3	RES	Restore
DC4	Device Control 4	RFF	Required Form Feed
DEL	Delete	RNL	Required New Line
DLE	Data Link Escape	RPT	Repeat
DS	Digit Select	SA	Set Attribute
EM	End of Medium	SBS	Subscript
ENP	Enable Presentation	SEL	Select
ENQ	Enquiry	SFE	Start Field Extended
EO	Eight Ones	SI	Shift In
EOT	End of Transmission	SM	Set Mode
ESC	Escape	SO	Shift Out
ETB	End of Transmission Block	SOH	Start of Heading
ETX	End of Text	SOS	Start of Significance
FF	Form Feed	SPS	Superscript
FS	Field Separator	STX	Start of Text
GE	Graphic Escape	SUB	Substitute
HT	Horizontal Tab	SW	Switch
IFS	Interchange File Separator	SYN	Synchronous Idle
IGS	Interchange Group Separator	TRN	Transparent
INP	Inhibit Presentation	UBS	Unit Backspace
IR	Index Return	VT	Vertical Tab
IRS	Interchange Record Separator	WUS	Word Underscore

**Additional ISO-8 Control Character Representations**

APC	Application Program Command	PLD	Partial Line Down
BPH	Break Permitted Here	PLU	Partial Line Up
CCH	Cancel Character	PM	Privacy Message
CSI	Control Sequence Introducer	PU1	Private Use One
DCS	Device Control String	PU2	Private Use Two
ESA	End of Selected Area	SCI	Single Character Introducer
HTJ	Character Tabulation w/ Justification	SOS	Start of String
HTS	Character Tabulation Set	SPA	Start of Guarded Area
IFS	Information Separator Four	SSA	Start of Selected Area
IGS	Information Separator Three	SS2	Single Shift Two
IND	Index	SS3	Single Shift Three
IRS	Information Separator Two	ST	String Terminator
MW	Message Waiting	STS	Set Transmit State
NBH	No Break Here	US	Information Separator One
NEL	Next Line	VTS	Line Tabulation Set
OSC	Operating System Command		

**Formatting Character Representations**

NSP	Numeric Space	SP	Space
RSP	Required Space	SHY	Syllable Hyphen

**Two-Character BSC Data Link Controls**

Function	EBCDIC	ASCII
ACK-0	DLE,X'70'	DLE,0
ACK-1	DLE,X'61'	DLE,1
WACK	DLE,X'68'	DLE,;
RVI	DLE,X'7C'	DLE,<

**Commonly Used Editing Pattern Characters**

Code (Hex)	Meaning	Code (Hex)	Meaning
20	Digit selector	5B	Dollar sign
21	Start of significance	5C	Asterisk
22	Field separator	6B	Comma
40	Blank	C3D9	CR (credit)
4B	Period	C4C2	DB (debit)

## **ANSI-Defined Printer Control Characters**

(A in RECFM field of DCB)

Code	Action before Printing Record
blank	Space 1 line
0	Space 2 lines
-	Space 3 lines
+	Suppress space
1	Skip to line 1 on new page

## **Hexadecimal and Decimal Conversion**

*From hex:* locate each hex digit in its corresponding column position and note the decimal equivalents. Add these to obtain the decimal value.

*From decimal:* (1) locate the largest decimal value in the table that will fit into the decimal number to be converted, and (2) note its hex equivalent and hex column position. (3) Find the decimal remainder. Repeat the process on this and subsequent remainders.

**Note:** Hexadecimal equivalents of all numbers from 0 to 255 are listed in the code tables.

Word															
Halfword								Halfword							
Byte				Byte				Byte				Byte			
Bits:		0123		4567		0123		4567		0123		4567		0123	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	268,435,456	1	16,777,216	1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	536,870,912	2	33,554,432	2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	805,306,368	3	50,331,648	3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	1,073,741,824	4	67,108,864	4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	1,342,177,280	5	83,886,080	5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	1,610,612,736	6	100,663,296	6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	1,879,048,192	7	117,440,512	7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	2,147,483,648	8	134,217,728	8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	2,415,919,104	9	150,994,944	9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	2,684,354,560	A	167,772,160	A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	2,952,790,016	B	184,549,376	B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	3,221,225,472	C	201,326,592	C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	3,489,660,928	D	218,103,808	D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	3,758,096,384	E	234,881,024	E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	4,026,531,840	F	251,658,240	F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
	8		7		6		5		4		3		2		1

## Powers of 2 and 16

$m$	$n$	$2^m$ and $16^n$	Symbol
0	0	1	
1		2	
2		4	
3		8	
4	1	16	
5		32	
6		64	
7		128	
8	2	256	
9		512	
10		1 024	K (kilo)
11		2 048	
12	3	4 096	
13		8 192	
14		16 384	
15		32 768	
16	4	65 536	
17		131 072	
18		262 144	
19		524 288	
20	5	1 048 576	M (mega)
21		2 097 152	
22		4 194 304	
23		8 388 608	
24	6	16 777 216	
25		33 554 432	
26		67 108 864	
27		134 217 728	
28	7	268 435 456	
29		536 870 912	
30		1 073 741 824	G (giga)
31		2 147 483 648	
32	8	4 294 967 296	
33		8 589 934 592	
34		17 179 869 184	
35		34 359 738 368	
36	9	68 719 476 736	
37		137 438 953 472	
38		274 877 906 944	
39		549 755 813 888	
40	10	1 099 511 627 776	T (tera)
41		2 199 023 255 552	
42		4 398 046 511 104	
43		8 796 093 022 208	
44	11	17 592 186 044 416	
45		35 184 372 088 832	
46		70 368 744 177 664	
47		140 737 488 355 328	
48	12	281 474 976 710 656	
49		562 949 953 421 312	
50		1 125 899 906 842 624	P (peta)
51		2 251 799 813 685 248	
52	13	4 503 599 627 370 496	
53		9 007 199 254 740 992	
54		18 014 398 509 481 984	
55		36 028 797 018 963 968	
56	14	72 057 594 037 927 936	
57		144 115 188 075 855 872	
58		288 230 376 151 711 744	
59		576 460 752 303 423 488	
60	15	1 152 921 504 606 846 976	E (exa)
61		2 305 843 009 213 693 952	
62		4 611 686 018 427 387 904	
63		9 223 372 036 854 775 808	
64	16	18 446 744 073 709 551 616	







File Number: S-390-00



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SA22-7871-02

